

**Final Approach Spacing Tool
(FAST)
Scheduling Logic
Velocity Vector
Sensitivity Analysis**

K. Tysen Mueller

Prepared for:

National Aeronautics and Space Administration
Ames Research Center
Moffett Field, CA 94035-1000

Under Subcontract to:

Honeywell Inc.
Houston Engineering Center
Houston, TX 77058

Honeywell Subcontract No. 8021308
CLIN 006, Task Order 15
NASA Prime Contract No. NAS2-98001

Seagull Project No. C175.15

13 November 1998

Table of Contents

| | | |
|-----|---|----|
| 1.0 | EXECUTIVE SUMMARY | |
| 1.1 | Background | 11 |
| 1.2 | Report Outline | 12 |
| 1.3 | Key Results | 13 |
| 1.4 | Conclusions and Recommendations | 16 |
| 2.0 | FAST SCHEDULING LOGIC | |
| 2.1 | Introduction | 19 |
| 2.2 | Nominal Scheduling Logic | 19 |
| 2.3 | Proposition Inputs | 25 |
| | 2.3.1 Normalized Separation Distance at FCTS | 26 |
| | 2.3.2 Normalized Separation Distance at Current Position | 26 |
| | 2.3.3 Normalized Separation Distance along Downwind Segment | 27 |
| | 2.3.4 Horizontal Separation | 27 |
| | 2.3.5 Altitude Separation | 28 |
| | 2.3.6 Relative Ground Speed | 28 |
| | 2.3.7 Relative ETA Magnitude | 28 |
| | 2.3.8 Controllability | 29 |
| | 2.3.9 Excess Delay | 30 |
| | 2.3.10 Normalized Delay Savings | 31 |
| 2.4 | Proposition Results | 31 |
| | 2.4.1 Relative Ground Speed | 32 |
| | 2.4.2 Normalized Separation Distance at FCTS | 34 |
| | 2.4.3 Relative ETA Magnitude | 37 |
| | 2.4.4 Excess Delay | 38 |
| | 2.4.5 Normalized Delay Savings (NDS) | 40 |
| | 2.4.6 Controllability (κ) | 41 |
| 3.0 | INPUT VARIABLE ERROR ANALYSIS | |
| 3.1 | Introduction | 45 |
| 3.2 | Nominal TRACON Trajectories | 46 |
| 3.3 | Input Variable Estimation Errors | 50 |
| | 3.3.1 Aircraft Tracking Errors | 51 |
| | 3.4.2 Relative Ground Speed | 55 |
| | 3.4.3 Normalized Separation Distance at FCTS | 57 |
| | 3.4.4 Relative ETA Magnitude | 58 |
| | 3.4.5 Controllability | 61 |
| | 3.4.6 Excess Delay | 61 |
| | 3.4.7 Normalized Delay Savings | 65 |

Table of Contents -- continued

| | | |
|-------|---|-----|
| 4.0 | FIGURES OF MERIT | |
| 4.1 | Relative Ground Speed Figure of Merit | 67 |
| 4.2 | New Figures of Merit | 71 |
| 4.3 | Relative Ground Speed | 76 |
| 4.4 | Relative ETA Magnitude | 77 |
| 4.5 | NSD _{FCTS} | 79 |
| 4.6 | Excess Delay | 81 |
| 4.7 | Normalized Delay Savings | 83 |
| 4-8 | Controllability | 84 |
| 5.0 | ORDERING PROCEDURE PERFORMANCE SIMULATION RESULTS | |
| 5.1 | Ordering Procedure Performance Simulation | 87 |
| 5.2 | Nominal Trajectories | 89 |
| 5.3 | Ordering Procedure Input Error Statistics | 90 |
| 5.4 | Ordering Procedure Performance Statistics | 91 |
| 5.4.1 | Membership Statistics | 92 |
| 5.4.2 | Output Statistics | 94 |
| 5.4.3 | Weight Statistics | 95 |
| 5.4.4 | Firing Strength Statistics | 95 |
| 5.4.5 | Proposition 5 Results | 96 |
| 5.4.6 | Proposition 7 Results | 97 |
| 5.4.7 | Ordering Procedure Results | 98 |
| 6.0 | MERGING PROCEDURE PERFORMANCE SIMULATION RESULTS | |
| 6.1 | Merging Procedure Performance Simulation | 103 |
| 6.2 | Nominal Trajectories | 105 |
| 6.3 | Merging Procedure Input Statistics | 106 |
| 6.3.1 | NSD _{FCTS} Statistics | 106 |
| 6.3.2 | Relative Ground Speed Statistics | 107 |
| 6.4 | Merging Procedure Decision Statistics | 109 |
| 6.4.1 | Membership Statistics | 109 |
| 6.4.2 | Output Statistics | 109 |
| 6.4.3 | Weight Statistics | 109 |
| 6.4.4 | Firing Strength Statistics | 110 |
| 6.4.5 | Proposition 1 - 3 Results | 112 |
| 6.4.6 | Proposition 5 and 7 Results | 113 |
| 6.4.7 | Merging Procedure Results | 113 |
| 7.0 | REFERENCES | 123 |

Table of Contents - continued

APPENDICES

| | |
|--|----------------------------------|
| A: FAST TS MATLAB SIMULATION | |
| A.1 | Overview 3 |
| A.2 | Test Case 4 |
| A.3 | fastTS.m MATLAB Listing 9 |
| A.4 | fastTSin.m MATLAB Listing 21 |
| A.5 | TSplotfunc.m MATLAB Listing 25 |
| B: ORDERING PROCEDURE PERFORMANCE SIMULATION | |
| B.1 | Overview 27 |
| B.2 | Test Case 28 |
| B.3 | fastSL1.m MATLAB Listing 35 |
| B.4 | fastSL1inpar.m MATLAB Listing 40 |
| B.5 | fastSL1in.m MATLAB Listing 42 |
| B.6 | abfltrfunc.m MATLAB Listing 46 |
| B.7 | vgfunc.m MATLAB Listing 49 |
| B.8 | symbrfunc.m MATLAB Listing 51 |
| B.9 | vgscrfunc.m MATLAB Listing 53 |
| B.10 | mbrfunc.m MATLAB Listing 55 |
| B.11 | SL1plotfunc.m MATLAB Listing 57 |
| C: MERGING PROCEDURE PERFORMANCE SIMULATION | |
| C.1 | Overview 67 |
| C.2 | Test Case 68 |
| C.3 | fastSL2.m MATLAB Listing 77 |
| C.4 | fastSL2inpar.m MATLAB Listing 83 |
| C.5 | fastSL2in.m MATLAB Listing 85 |
| C.6 | intfunc.m MATLAB Listing 90 |
| C.7 | fctscrfunc.m MATLAB Listing 91 |
| C.8 | SL2plotfunc.m MATLAB Listing 94 |

Figures

| | | |
|------|---|----|
| 1-1 | Final Approach Spacing Tool (FAST) Information Flow | 11 |
| 1-2 | Approach to Dallas-Ft. Worth Runway 18R from SW Metering Fix | 12 |
| 1-3 | FAST SL Performance Simulation Information Flow | 14 |
| 2-1 | Relative Ground Speed Membership and Consequent Functions | 32 |
| 2-2 | NSD _{FCTS} Propositions | 35 |
| 2-3 | Relative ETA Magnitude Proposition | 37 |
| 2-4 | 'B Sequenced Behind A' Excess Delay Propositions | 38 |
| 2-5 | 'A Sequenced Behind B' Excess Delay Propositions | 39 |
| 2-6 | Normalized Delay Savings Propostions | 41 |
| 2-7 | Controllability Propositions (A Sequenced Behind B) | 42 |
| 2-8 | Controllability Propositions (B Sequenced Behind A) | 43 |
| 3-1 | Aircraft Radar Tracks to Dallas-Ft. Worth Runway 18R from SW Metering Fix (11 July 1996) | 47 |
| 3-2 | Approach to Dallas-Ft. Worth Runway 18R from SW Metering Fix | 47 |
| 3-3 | FAST TS Jet and Turboprop Distance Histories | 48 |
| 3-4 | FAST TS Jet and Turboprop Time Histories | 49 |
| 3-5 | FAST TS Jet 1 Ahead of Jet 2 Time Histories | 50 |
| 3-6 | FAST TS Turboprop Ahead of Jet Time Histories | 51 |
| 3-7 | Jet Ground Speed Tracking Errors | 52 |
| 3-8 | Turboprop Ground Speed Tracking Errors | 53 |
| 3-9 | Jet Heading Tracking Errors | 53 |
| 3-10 | Turboprop Heading Tracking Errors | 54 |
| 3-11 | Jet Ground Path Tracking Errors | 54 |
| 3-12 | Turboprop Ground Path Tracking Errors | 55 |
| 3-13 | Relative Ground Speed Statistics (Jet 1 ahead of Jet 2) | 56 |
| 3-14 | Relative Ground Speed at FCTS Statistics (Turboprop ahead of Jet) | 56 |
| 3-15 | NSD at FCTS Statistics (Turboprop ahead of Jet) | 59 |
| 3-16 | Relative ETA Magnitude Statistics (Jet 1 ahead of Jet 2) | 60 |
| 3-17 | Controllability Statistics (Jet 1 ahead of Jet 2) | 62 |
| 3-18 | Excess Delay Statistics (Jet 1 ahead of Jet 2) | 64 |
| 3-19 | Normalized Delay Savings Statistics (Jet 1 ahead of Jet 2) | 66 |
| 4-1 | Relative Ground Speed Membership and Consequent Functions | 67 |
| 4-2 | Figure of Merit for Relative Ground Speed (Case 1) | 68 |
| 4-3 | Figure of Merit for Relative Ground Speed (Case 2) | 69 |
| 4-4 | Figure of Merit for Relative Ground Speed (Case 3) | 69 |
| 5-1 | FAST SL Ordering Procedure Performance Simulation | 88 |
| 5-2 | FAST TS Jet 1 and Jet 2 Time Histories | 90 |
| 5-3 | Relative Ground Speed Statistics (Jet 1 ahead of Jet 2) | 91 |
| 5-4 | Ordering Procedure Input Histories | 92 |
| 5-5 | Ordering Procedure Membership Histories (Jet 1 ahead of Jet 2) | 93 |
| 5-6 | Ordering Procedure Output Histories (Jet 1 ahead of Jet 2) | 94 |
| 5-7 | Ordering Procedure Weight Histories (Jet 1 ahead of Jet 2) | 96 |
| 5-8 | Ordering Procedure Firing Strength Histories (Jet 1 ahead of Jet 2) | 97 |

Figures - continued

| | |
|-----|---|
| 5-9 | Ordering Procedure Normalized Firing Strength Histories |
|-----|---|

| | | |
|------|--|-----|
| | (Jet 1 ahead of Jet 2) | 98 |
| 5-10 | 'Is Faster' Proposition Results (Jet 1 ahead of Jet 2) | 99 |
| 5-11 | 'Is Close AND Faster' Proposition Results (Jet 1 ahead of Jet 2) | 100 |
| 5-12 | Ordering Procedure Results (Jet 1 ahead of Jet 2) | 101 |
| 6-1 | FAST SL Merging Procedure Performance Simulation | 104 |
| 6-2 | FAST TS Time Histories (Turboprop ahead of Jet) | 106 |
| 6-3 | NSD _{FCTS} Statistics (Turboprop ahead of Jet) | 107 |
| 6-4 | Relative Ground Speed Statistics (Turboprop ahead of Jet) | 108 |
| 6-5 | Merging Procedure Input Histories (Turboprop ahead of Jet) | 110 |
| 6-6 | Merging Procedure Membership Histories (Turboprop ahead of Jet) | 111 |
| 6-7 | Merging Procedure Output Histories (Turboprop ahead of Jet) | 112 |
| 6-8 | Merging Procedure Weight Histories (Turboprop ahead of Jet) | 113 |
| 6-9 | Merging Procedure Firing Strength Histories (Turboprop ahead of Jet) | 114 |
| 6-10 | Merging Procedure Normalized Firing Strength Histories (Turboprop ahead of Jet) | 115 |
| 6-11 | 'Is Significantly Delayed' Proposition Results (Turboprop ahead of Jet) | 116 |
| 6-12 | 'Is Delayed' Proposition Results (Turboprop ahead of Jet) | 117 |
| 6-13 | 'Is Slightly Delayed' Proposition Results (Turboprop ahead of Jet) | 118 |
| 6-14 | 'Is Faster' Proposition Results (Turboprop ahead of Jet) | 119 |
| 6-15 | 'Is Close' AND 'Is Faster at FCTS' Proposition Results (Turboprop ahead of Jet) | 120 |
| 6-16 | Merging Procedure Results (Turboprop ahead of Jet) | 121 |

Appendices

| | | |
|-----|---|----|
| A-1 | fastTS.m Test Case Plot 1 (TFLAG = 1) | 6 |
| A-2 | fastTS.m Test Case Plot 2 (TFLAG = 1) | 7 |
| A-3 | fastTS.m Test Case Plot 3 (TFLAG = 1) | 7 |
| A-4 | fastTS.m Test Case Plot 4 (TFLAG = 1) | 8 |
| B-1 | FAST SL Ordering Procedure Performance Simulation | 27 |
| B-2 | Nominal Relative Variables (Jet 1 - Jet 2) | 29 |
| B-3 | Relative Ground Speed Statistics (Jet 1 - Jet 2) | 29 |
| B-4 | Ordering Procedure Nominal and Statistical Results (Jet 1 and Jet 2) | 30 |
| B-5 | 'Is Faster' Proposition Nominal and Statistical Results (Jet 1 and Jet 2) | 30 |
| B-6 | 'Is Closer and Faster' Proposition Nominal and Statistical Results (Jet 1 and Jet 2) | 31 |
| B-7 | Proposition Nominal and Statistical Firing Strength Results (Jet 1 - Jet 2) | 31 |
| B-8 | Proposition Nominal and Statistical Normalized Firing Strength Results (Jet 1 - Jet 2) | 32 |
| B-9 | Proposition Nominal and Statistical Input Variables (Jet 1 - Jet 2) | 32 |

Figures - continued

| | | |
|------|--|----|
| B-10 | Proposition Nominal and Statistical Membership Values (Jet 1 - Jet 2) | 33 |
| B-11 | Proposition Nominal and Statistical Output Values (Jet 1 - Jet 2) | 33 |
| B-12 | Proposition Nominal and Statistical Weight (Jet 1 - Jet 2) | 34 |

| | | |
|------|--|----|
| C-1 | FAST SL Merging Procedure Performance Simulation | 67 |
| C-2 | Nominal Relative Variables (Tprop - Jet) | 69 |
| C-3 | NSD _{FCTS} Statistics (Tprop - Jet) | 69 |
| C-4 | Relative Ground Speed Statistics (Tprop - Jet) | 70 |
| C-5 | Merging Procedure Nominal and Statistical Results (Tprop - Jet) | 70 |
| C-6 | 'Is Significantly Delayed' Proposition Nominal and Statistical Results (Tprop - Jet) | 71 |
| C-7 | 'Is Delayed' Proposition Nominal and Statistical Results (Tprop - Jet) | 71 |
| C-8 | 'Is Slightly Delayed' Proposition Nominal and Statistical Results (Tprop - Jet) | 72 |
| C-9 | 'Is Faster' Proposition Nominal and Statistical Results (Tprop - Jet) | 72 |
| C-10 | 'Is Closer and Faster' Proposition Nominal and Statistical Results (Tprop - Jet) | 73 |
| C-11 | Proposition Nominal and Statistical Firing Strength Results (Tprop - Jet) | 73 |
| C-12 | Proposition Nominal and Statistical Normalized Firing Strength Results (Tprop - Jet) | 74 |
| C-13 | Proposition Nominal and Statistical Input Variables (Tprop - Jet) | 74 |
| C-14 | Proposition Nominal and Statistical Membership Values (Tprop - Jet) | 75 |
| C-15 | Proposition Nominal and Statistical Output Values (Tprop - Jet) | 75 |
| C-16 | Proposition Nominal and Statistical Weight (Tprop - Jet) | 76 |

Tables

| | | |
|------|--|-----|
| 2-1 | Ordering Procedure of a GENERAL-Type Spatial Constraint | 20 |
| 2-2 | Merging Procedure of a GENERAL-Type Spatial Constraint | 21 |
| 2-3 | Ordering Procedure of a FINAL-Type Spatial Constraint | 22 |
| 2-4 | Merging Procedure of a FINAL -Type Spatial Constraint | 23 |
| 2-5 | Proposition Output Values | 25 |
| 2-6 | Aircraft Minimum Required Separation Distance, $\Delta d_{A,B}$ (nmi.) | 26 |
| 2-7 | Aircraft Minimum Relative Time Spacing, $\Delta t_{A,B}$ (min.) | 30 |
| 2-8 | NSD_{FCTS} Proposition Parameters | 37 |
| 2-9 | Relative ETA Magnitude Proposition Parameters | 37 |
| 2-10 | Excess Delay Proposition Parameters (Aircraft A Sequenced Behind B) | 40 |
| 2-11 | Excess Delay Proposition Parameters (Aircraft B Sequenced Behind A) | 40 |
| 2-12 | NDS Proposition Parameters | 40 |
| 2-13 | Controllability Proposition Parameters | 42 |
| 3-1 | Input Variable Dependence on Ground Speed | 45 |
| 3-2 | FAST SL Ordering/Merging Procedure Dependence on Relative Ground Speed, ΔV_G | 46 |
| 4-1 | Relative Ground Speed Decision Error Cases | 68 |
| 4-2 | ΔV_G Figure of Merit Parameters | 70 |
| 4-3 | Relative Ground Speed Proposition Parameters | 76 |
| 4-4 | NSD_{FCTS} Proposition Parameters | 79 |
| 4-5 | SD_{FCTS} Proposition Parameters | 80 |
| 4-6 | Excess Delay Proposition Parameters (Aircraft A Sequenced Behind B) | 81 |
| 4-7 | Excess Delay Proposition Parameters (Aircraft B Sequenced Behind A) | 81 |
| 4-8 | NDS Proposition Parameters | 83 |
| 4-9 | Controllability Proposition Parameters | 84 |
| 5-1 | Ordering Procedure of a GENERAL-Type Spatial Constraint | 87 |
| 6-1 | Merging Procedure of a GENERAL-Type Spatial Constraint | 103 |

Appendices

| | | |
|-----|--|----|
| A-1 | fastTS.m Test Case MATLAB Screen Capture (TFLAG = 1) | 5 |
| A-2 | fastTSjet.dat Test Case Data File Abstract (TFLAG = 1) | 8 |
| A-3 | fastTSjet.dat Test Case Data File Format | 9 |
| B-1 | fastSL1stats.dat Statistics Data File Abstract | 34 |
| B-2 | fastSL1stats.dat File Format | 35 |
| C-1 | fastSL2stats.dat Statistics Data File Abstract | 76 |
| C-2 | fastSL2stats.dat File Format | 77 |

1.0 EXECUTIVE SUMMARY

1.1 Background

This report presents the error models and figures of merit to evaluate the performance of the Passive Final Approach Spacing Tool (FAST). Specifically, the ground speed radar tracking errors which are introduced into the FAST software are investigated. The error models describing the effect of these ground speed tracking errors on the FAST Scheduling Logic (SL) are developed. These error models and a number of figures of merit are then integrated into two FAST Performance Simulations. These Performance Simulations are then evaluated for typical two-aircraft scenarios. It is shown that there is an increasing likelihood that FAST SL might reach an incorrect aircraft ordering or merging decision due to ground speed tracking errors. However, the FAST SL logic is in general relatively robust to these types of errors.

Passive FAST provides advisories and serves as a decision aid to the air traffic controller (Anon, 1990; Bergh, 1995; Davis, 1997; Erzberger, 1995; Lee, 1995; Mueller, 1998; Robinson, 1997; Slattery, 1995, 1997). As illustrated in Figure 1-1, the process starts with estimates of the aircraft horizontal position and velocity as obtained from the ARTS radar and its tracking software. Next altitude information is added, as provided by the aircraft barometric altimeter. FAST then computes predicted trajectories for the aircraft to the runway threshold with the Trajectory Synthesizer (TS) using estimates of the aircraft's current position and velocity. FAST then determines the preferred sequencing or merging of aircraft in the Terminal Radar Area Control (TRACON) using FAST SL, based on logic which mimics the air traffic controller's decision process.

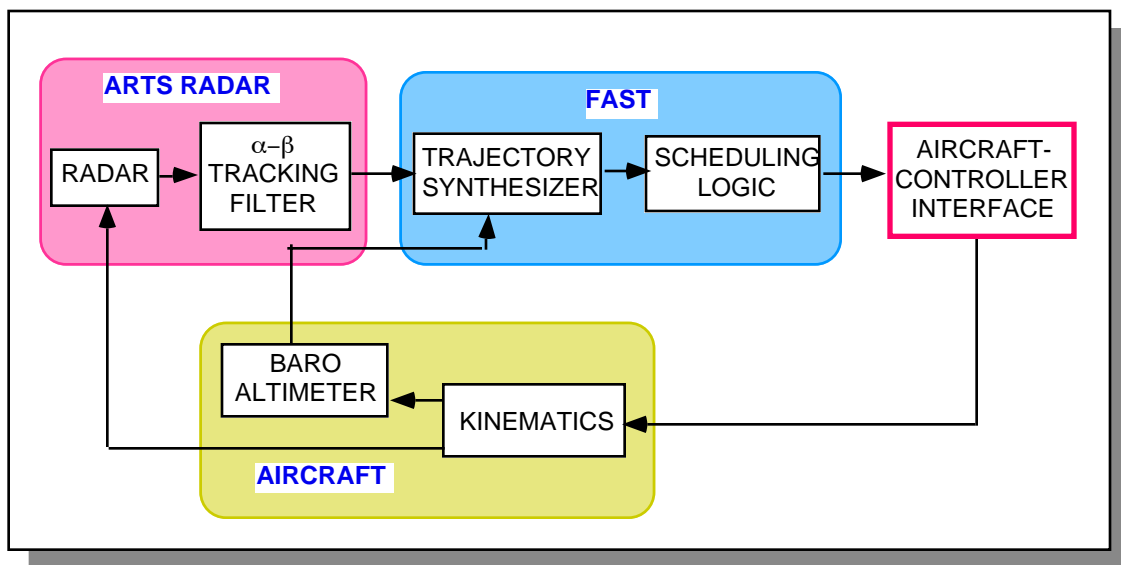


Figure 1-1. Final Approach Spacing Tool (FAST) Information Flow
Figure 1-2 illustrates the typical aircraft flight path segments in the TRACON terminal approach area. This figure illustrates the problem of ordering aircraft on the same flight path segment and merging them onto a common flight path segment. The aircraft are spaced and merged to assure that:

- minimum separation constraints are maintained
- unwanted overtakes are avoided
- delays in reaching the runway are minimized.

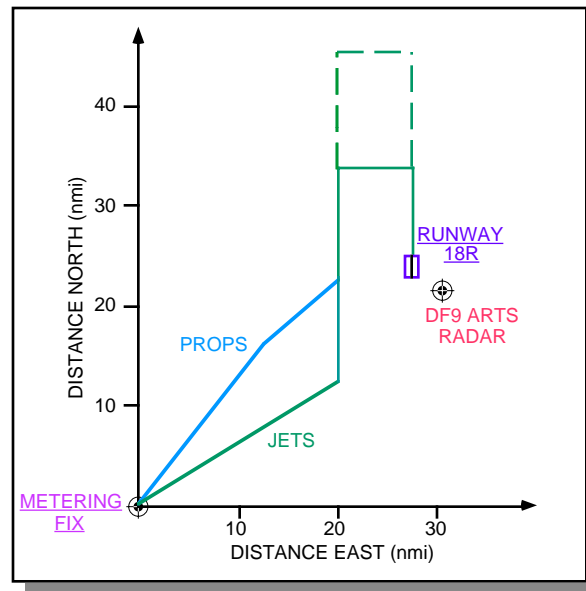


Figure 1-2. Approach to Dallas-Ft. Worth Runway 18R from SW Metering Fix

Under a prior study (Mueller, 1998), the focus was primarily on the sources of the velocity errors and their impact on the FAST TS. The study also evaluated a few of the input variables into the FAST SL which are sensitive to velocity errors.

It was the objective of this study to complete the error analysis for the remaining FAST SL input variables which are sensitive to velocity errors. This was followed with the development of two statistical simulations of the FAST SL fuzzy logic. These performance simulations model the individual FAST SL input errors and several figures of merit. These statistical error models were then used to evaluate the performance of FAST over several nominal aircraft flight path profiles.

1.2 Report Outline

Chapter 2 introduces the four Procedures used by the FAST SL, two of which are the focus of this study. Also, this chapter presents the mathematical definitions for all ten Proposition input variables used by these four Procedures. In addition the mathematical models for the nominal Membership function, Output, Weight, and Firing Strength are presented for the six Proposition inputs which are ground speed dependent.

Chapter 3 focuses on the input variable statistics. It specifically shows the nominal trajectories which are used to drive the input variable error models developed in this

Chapter. The input variable statistics for the six ground speed dependent input variables are then simulated using nominal trajectories.

Chapter 4 introduces and examines various figures of merit for evaluating the performance of the FAST SL. This Chapter includes a discussion of the figure of merit developed under the last study and then introduces four new figures of merit. These new figures of merit are shown to provide a much more practical metric for evaluating the FAST SL performance.

Chapter 5 summarizes the FAST SL Ordering Procedure Performance Simulation and presents results for a two-aircraft scenario. Chapter 6, in turn, summarizes the FAST SL Merging Procedure Performance Simulation and presents results for another two-aircraft scenario.

Appendix A presents the FAST TS MATLAB simulation listing and test case, developed under the last study. Appendix B presents the FAST SL Ordering Procedure Performance Simulation listing and test case while Appendix C presents the FAST SL Merging Procedure Performance Simulation listing and test case.

1.3 Key Results

The two performance simulations which were developed are illustrated in Figure 1-3. This figure shows that one simulation is used to evaluate the nominal and statistical performance of the Ordering Procedure of a GENERAL-Type Spatial Constraint. A second simulation is used to evaluate the nominal and statistical performance of the Merging Procedure of a GENERAL-Type Spatial Constraint.

As shown in this figure, each performance simulation takes the nominal trajectories of two aircraft as input. It then computes the radar tracking ground speed errors and the corresponding trajectory prediction errors. With the trajectory prediction errors, the individual FAST SL Propositions are evaluated using both the nominal and the statistical inputs. The performance figures of merit are evaluated for each Proposition in the Procedure as well as for the composite Procedure.

For these two scenarios, the Relative Ground Speed mean errors were found to vary between $-/+25$ knots while the standard deviation was less than 7 knots. While a negative Relative Ground Speed could lead to an overtake situation, the current deadband used by the Relative Ground Speed Proposition is $-/+20$ knots. This suggests that the Relative Ground Speed errors are large enough to introduce Proposition decision errors. Alternately, this suggests that the dead band might be set slightly higher.

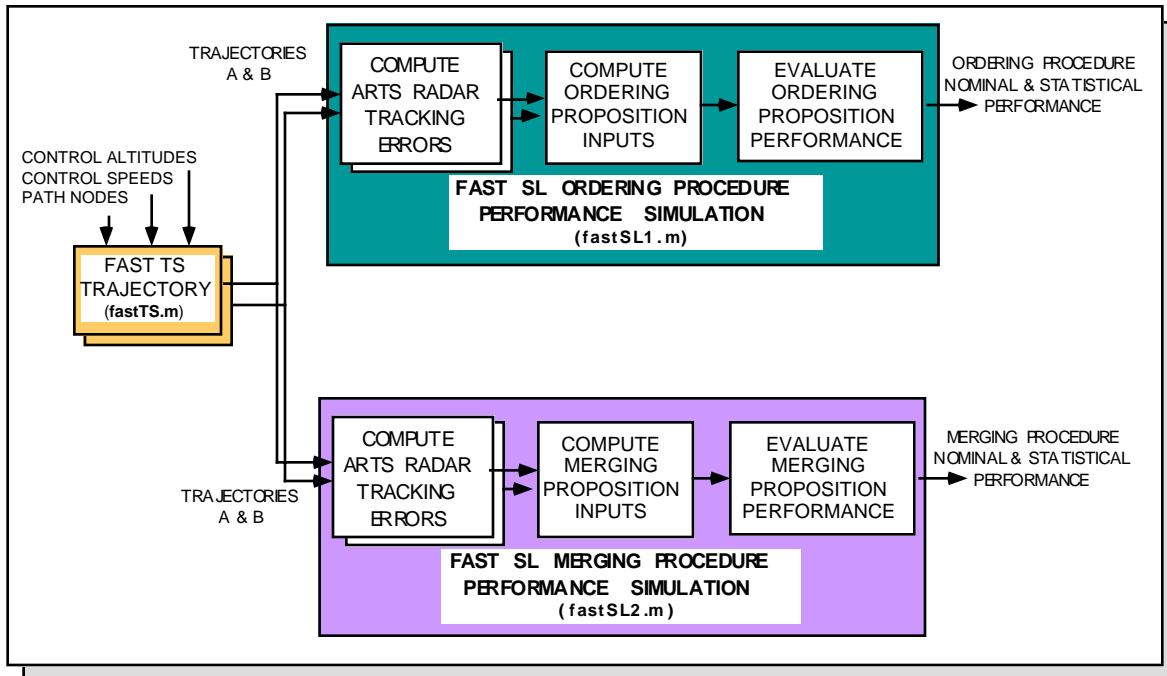


Figure 1-3. FAST SL Performance Simulation Information Flow

The mean error of NSD_{FCTS} for two aircraft merging onto the Downwind segment varied between -1 and 0 while the standard deviation was less than 0.25. Since the nominal NSD_{FCTS} is 1, the estimate (sum of nominal and mean) varied between 0 and 1. Since an NSD_{FCTS} of 1 indicates a filled aircraft slot, an estimate of $NSD_{FCTS} < 1$ suggests the loss of a slot or alternately a minimum separation conflict between the two aircraft. This perceived conflict can result in an unneeded additional aircraft separation request by the air traffic controller.

For the time of arrival dependent variables, the Relative Estimated Time of Arrival (ETA) Magnitude mean error varied between $-/+1$ minute while the standard deviation was less than 0.2 minutes. The nominal and minimum allowed time separation between two jets is 1 minute for this scenario. Hence, the Relative ETA Magnitude estimate could lead to a perceived violation of this requirement. Alternately, if this requirement is actually being violated, the ETA magnitude statistics can mask this problem. This would lead to the conclusion that minimum time separation standards are being observed.

The Controllability is another ETA-dependent input variable. Its mean error varied between $-/+2$ while the standard deviation was less than 0.3. The Excess Delay mean error, another ETA-dependent input variable, varied between $-/+5\%$ while the standard deviation was less than 2%. Finally, the Normalized Delay Savings (NDS) mean, the fourth ETA-dependent input variable, varied between $-/+0.2$ while the standard deviation was less than 0.3.

The last study selected a performance figure of merit which focused on the statistics of the Proposition input variables. This figure of merit was defined as the probability that an estimate of the input to a trapezoidal Membership function is located in a different

region (within the Deadband, or to the right or left of the Deadband) than the nominal input. This leads to a metric which determines the probability of a Proposition decision error -- somewhat analogous to missed and false alarms in signal theory.

Under this study, it was determined that a more significant figure of merit is the mean (expected) Proposition Firing Strength. The Proposition Firing Strength determines the decision reached by that Proposition. The nominal Firing Strength is defined as the product of the nominal Proposition Output and the Weight. Both are directly determined by the nominal Proposition Membership Function and the input to it. Hence, the mean Membership, Output, and Weight were also added as figures of merit.

For the Ordering Procedure of a GENERAL-Type Spatial Constraint, the only ground speed dependent input is the Relative Ground Speed itself. This input is used in two of the seven Proposition pairs. For a scenario consisting of the two in-track jet aircraft, it was shown that the Relative Ground Speed statistics coupled with the structure of the two Propositions introduced only small to moderate deviations from the nominal firing strengths. When these Proposition statistics were combined with the remaining nominal Proposition results, the Procedure Firing Strength statistics also showed only minor deviations from the nominal Procedure results. Caution must be used in generalizing these results since they were obtained from only a single two-aircraft scenario.

For the Merging Procedure of a GENERAL-Type Spatial Constraint, both the Relative Ground Speed and the NSD_{FCTS} of the two aircraft are dependent on ground speed errors. Furthermore, the NSD_{FCTS} input is used in three of the seven Proposition pairs. Also, their possible Output values are much more significant than those used for the Relative Ground Speed Propositions. In addition, the Relative Ground Speed was used as input in two of the seven Propositions.

For a scenario of one turboprop aircraft merging onto the same Downwind flight path segment as a jet aircraft, the Firing Strength statistics for the NSD_{FCTS} Propositions showed small to large deviations from the nominal results. The Relative Ground Speed Firing Strength Statistics were shown to produce small deviations from the nominal for the two associated Propositions. When the Firing Strength statistics from all seven Propositions were combined to obtain the Procedure Firing Strength statistics, it was found that Procedure Firing Strength statistics showed some moderate deviations from the nominal Procedure results. However, the results were not significant enough to lead to an erroneous merging order decision for the two aircraft. In other words, the polarity of the Procedure Firing Strength did not change sign. Again, caution should be used in generalizing these results since they are only based on one scenario.

1.4 Conclusions and Recommendations

Under this study, the remaining error models were developed which allow all six of the ten Proposition input variables which are ground speed dependent to be evaluated. These error models were incorporated into a performance simulation of the Ordering Procedure of a GENERAL-Type Spatial Constraint and into a separate simulation of the

Merging Procedure of a GENERAL-Type Spatial Constraint. In addition, four new figures of merit were defined and incorporated into these two simulations.

The results obtained from a two-aircraft scenario with the Ordering Procedure performance simulation, showed that the ground speed dependent input variables had negligible effect on the Procedure decision for this scenario. The results obtained for a second two-aircraft scenario with the Merging Procedure performance simulation showed a moderate impact on the Procedure decision for this scenario. However, the impact was not severe enough to result in an incorrect Procedure decision. Caution however should be used in generalizing these results since they are based on one scenario for each Procedure.

Overall, two performance simulations have been developed which can evaluate the performance of two of the four Procedures used by FAST. When the performance simulations for the last two Procedures are developed, the complete performance of FAST SL can be determined for all four Procedures.

It is recommended that further work on Passive FAST be performed in the following areas:

- Complete FAST SL Performance Simulations -- include FINAL Procedures
- Perform a number of FAST SL Simulation runs -- understand how well FAST works under simulated scenarios of a typical airport
- Explore FAST performance under Free-Flight -- availability of more accurate ground speed data as well as aircraft maneuver intent information

While the focus of this and the last study has been on the performance of Passive FAST, the error models and various simulations developed under both studies can be extended to Active FAST. With these extended tools, it will be possible to provide design tools for expediting the design of Active FAST. This follows from the fact that these extended tools will provide direct and efficient feedback of how well the proposed Active FAST architecture will perform under a variety of different simulated scenarios.

The tasks which are recommended in support of Active FAST are:

- Enhance FAST TS simulation -- incorporate FAST SL specified speed and heading maneuvers
- Develop generic nominal FAST SL simulation -- aid design process
- Develop generic statistical FAST SL simulation -- determine sensitivity to relative ground speed errors
- Explore design and performance impact of flight technical errors, wind errors, and unexpected weather fronts -- develop more robust FAST

2.0 FAST SCHEDULING LOGIC

2.1 Introduction

In this chapter, a general overview of the FAST SL fuzzy logic is presented. It also includes a description of the inputs used by this logic and the decisions which result from these inputs. A more thorough description of FAST SL can be found in (Mueller, 1998; Robinson, 1997).

The FAST Scheduling Logic (SL) sequences and merges all aircraft approaching the same runway, such that the maximum number of flights can be safely landed. To assure that this scheduling is performed safely, FAST SL checks the minimum current and future predicted aircraft separations to assure that these are not less than the specified minimums nor excessively large. FAST SL also checks the current and future predicted relative speeds of all aircraft on the same current or future flight path segment to avoid unwanted overtakes. Finally, FAST SL attempts to schedule the aircraft such that there is minimum delay in each aircraft reaching its assigned runway. It accomplishes this by evaluating the current position and velocity and future predicted positions, velocities, and ETAs for all aircraft in the TRACON.

FAST SL incorporates a fuzzy logic decision methodology to perform aircraft scheduling. The design of this logic has evolved over a number of years, incorporating inputs from air traffic controllers, and is still undergoing changes. As a result, the fuzzy logic decision methodology attempts to closely mimic the decision process the air traffic controllers themselves use in scheduling aircraft. Despite this capability, FAST is a decision aid to the controller rather than an automation of his duties.

2.2 Nominal Scheduling Logic

There are four primary sets of scheduling decisions which are evaluated in FAST using a set of fuzzy logic Procedures. These are:

- Ordering Procedure of a GENERAL-type Spatial Constraint
- Merging Procedure of a GENERAL-type Spatial Constraint
- Ordering Procedure for a FINAL-type Spatial Constraint
- Merging Procedure for a FINAL-type Spatial Constraint

A Spatial Constraint, or Spatial Constraint Set, is any group of aircraft whose trajectories currently, or in the future, will pass through the same section of airspace. A GENERAL-type Spatial Constraint is a set of aircraft from a particular trajectory segment, other than the Final segment.

The Propositions for the Ordering Procedure of a GENERAL-type Spatial Constraint are summarized in Table 2-1 for two aircraft, A and B. The Propositions for the Merging Procedure of a GENERAL-type Spatial Constraint are summarized in Table 2-2 for two aircraft. In Table 2-2, FCTS is the First Common Time Step of two aircraft which are merging onto a common flight segment, such as the Downwind segment.

Table 2-1 Ordering Procedure of a GENERAL-Type Spatial Constraint

| Number | Proposition | Input | Consequent | Output |
|--------|--|---------------------------|--------------------------------|--------|
| 1 | A <u>significantly ahead</u> of B <u>at current position</u> | NSD_{TRK} | A <u>significantly favored</u> | 45 |
| 2 | A <u>ahead</u> of B <u>at current position</u> | NSD_{TRK} | A <u>favored</u> | 30 |
| 3 | A <u>slightly ahead</u> of B <u>at current position</u> | NSD_{TRK} | A <u>slightly favored</u> | 15 |
| 4 | A <u>ahead</u> of B <u>at current position</u> | NSD_{TRK} | A <u>slightly favored</u> | 15 |
| 5 | A <u>faster</u> than B <u>at current position</u> | ΔV_G | A <u>marginally favored</u> | 7.5 |
| 6 | A <u>lower</u> than B <u>at current position</u> | Δh | A <u>marginally favored</u> | 7.5 |
| 7 | (A <u>close</u> to B) AND (A <u>faster</u> than B <u>at current position</u>) | Δd & ΔV_G | A <u>slightly favored</u> | 15 |
| 8 | B <u>significantly ahead</u> of A <u>at current position</u> | NSD_{TRK} | B <u>significantly favored</u> | -45 |
| 9 | B <u>ahead</u> of A <u>at current position</u> | NSD_{TRK} | B <u>favored</u> | -30 |
| 10 | B <u>slightly ahead</u> of A <u>at current position</u> | NSD_{TRK} | B <u>slightly favored</u> | -15 |
| 11 | B <u>ahead</u> of A <u>at current position</u> | NSD_{TRK} | B <u>slightly favored</u> | -15 |
| 12 | B <u>faster</u> than A <u>at current position</u> | ΔV_G | B <u>marginally favored</u> | -7.5 |
| 13 | B <u>lower</u> than A <u>at current position</u> | Δh | B <u>marginally favored</u> | -7.5 |
| 14 | (B <u>close</u> to A) AND (B <u>faster</u> than A <u>at current position</u>) | Δd & ΔV_G | B <u>slightly favored</u> | -15 |

Table 2-2 Merging Procedure of a GENERAL-Type Spatial Constraint

| Number | Proposition | Input | Consequent | Output |
|--------|--|---------------------------|--------------------------------|--------|
| 1 | A <u>significantly ahead</u> of B <u>at FCTS</u> | NSD_{FCTS} | A <u>significantly favored</u> | 45 |
| 2 | A <u>ahead</u> of B <u>at FCTS</u> | NSD_{FCTS} | A <u>favored</u> | 30 |
| 3 | A <u>slightly ahead</u> of B <u>at FCTS</u> | NSD_{FCTS} | A <u>slightly favored</u> | 15 |
| 4 | A <u>ahead</u> of B <u>at current position</u> | NSD_{TRK} | A <u>slightly favored</u> | 15 |
| 5 | A <u>faster</u> than B <u>at FCTS</u> | ΔV_G | A <u>marginally favored</u> | 7.5 |
| 6 | A <u>lower</u> than B <u>at current position</u> | Δh | A <u>marginally favored</u> | 7.5 |
| 7 | (A <u>close</u> to B) AND (A <u>faster</u> than B <u>at FCTS</u>) | Δd & ΔV_G | A <u>slightly favored</u> | 15 |
| 8 | B <u>significantly ahead</u> of A <u>at FCTS</u> | NSD_{FCTS} | B <u>significantly favored</u> | -45 |
| 9 | B <u>ahead</u> of A <u>at FCTS</u> | NSD_{FCTS} | B <u>favored</u> | -30 |
| 10 | B <u>slightly ahead</u> of A <u>at FCTS</u> | NSD_{FCTS} | B <u>slightly favored</u> | -15 |
| 11 | B <u>ahead</u> of A <u>at current position</u> | NSD_{TRK} | B <u>slightly favored</u> | -15 |
| 12 | B <u>faster</u> than A <u>at FCTS</u> | ΔV_G | B <u>marginally favored</u> | -7.5 |
| 13 | B <u>lower</u> than A <u>at current position</u> | Δh | B <u>marginally favored</u> | -7.5 |
| 14 | (B <u>close</u> to A) AND (B <u>faster</u> than A <u>at FCTS</u>) | Δd & ΔV_G | B <u>slightly favored</u> | -15 |

The Propositions for the Ordering Procedure of a FINAL-type Spatial Constraint are presented in Table 2-3. Finally, the Propositions for the Merging Procedure of a FINAL-type Spatial Constraint are summarized in Table 2-4. In this report, the focus will be on the Ordering and Merging Procedures of GENERAL-type Spatial Constraint.

As shown in these four tables, each Procedure incorporates a set of Propositions for a set of two aircraft (A and B). These Propositions typically use a relative dynamic variable, such as relative ground speed, relative separation, etc., as input. Each Proposition then determines a decision for these two aircraft. By combining all the Proposition decisions for a Procedure in a weighted sense, a Procedure decision is determined for these two aircraft. For all four Procedures, the decision which is reached

is that either the order of the two aircraft at the current, or a specified future time, is preferred or the reverse order is preferred.

Table 2-3 Ordering Procedure of a FINAL-Type Spatial Constraint

| Number | Proposition | Input | Consequent | Output |
|--------|---|--------------------------|--------------------------------|--------|
| 1 | A sequenced behind B is <u>significantly out of delay</u> | κ | A <u>significantly favored</u> | 45 |
| 2 | A sequenced behind B is <u>out of delay</u> | κ | A <u>favored</u> | 30 |
| 3 | (A was <u>ahead</u> of B) AND (B sequenced behind A is <u>not significantly delayed</u>) | $NSD_{TRK} \ \& \ \zeta$ | A <u>favored</u> | 30 |
| 4 | A sequenced behind B is <u>delayed</u> | ζ | A <u>slightly favored</u> | 15 |
| 5 | A sequenced behind B is <u>slightly delayed</u> | ζ | A <u>marginally favored</u> | 7.5 |
| 6 | A <u>ahead</u> of B along downwind segment | NSD_{DOWN} | A <u>favored</u> | 30 |
| 7 | A <u>lower</u> than B at current position | Δh | A <u>somewhat favored</u> | 22.5 |
| 8 | A sequenced ahead of B <u>causes less delay</u> | NDS | A <u>marginally favored</u> | 7.5 |
| 9 | B sequenced behind A is <u>significantly out of delay</u> | κ | B <u>significantly favored</u> | -45 |
| 10 | B sequenced behind A is <u>out of delay</u> | κ | B <u>favored</u> | -30 |
| 11 | (B was <u>ahead</u> of A) AND (A sequenced behind B is <u>not significantly delayed</u>) | $NSD_{TRK} \ \& \ \zeta$ | B <u>favored</u> | -30 |
| 12 | B sequenced behind A is <u>delayed</u> | ζ | B <u>slightly favored</u> | -15 |
| 13 | B sequenced behind A is <u>slightly delayed</u> | ζ | B <u>marginally favored</u> | -7.5 |
| 14 | B <u>ahead</u> of A along downwind segment | NSD_{DOWN} | B <u>favored</u> | -30 |
| 15 | B <u>lower</u> than A at current position | Δh | B <u>somewhat favored</u> | -22.5 |
| 16 | B sequenced ahead of A <u>causes less delay</u> | NDS | B <u>marginally favored</u> | -7.5 |

Table 2-4 Merging Procedure of a FINAL-Type Spatial Constraint

| Number | Proposition | Input | Consequent | Output |
|--------|---|-----------------------------|--------------------------------|--------|
| 1 | A sequenced behind B is <u>out of delay</u> | κ | A <u>significantly favored</u> | 45 |
| 2 | A sequenced behind B is <u>significantly delayed</u> | ζ | A <u>favored</u> | 30 |
| 3 | A sequenced behind B is <u>delayed</u> | ζ | A <u>slightly favored</u> | 15 |
| 4 | A sequenced behind B is <u>slightly delayed</u> | ζ | A <u>marginally favored</u> | 7.5 |
| 5 | A <u>ahead</u> of B <u>along downwind</u> | NSD_{DOWN} | A <u>somewhat favored</u> | 22.5 |
| 6 | A <u>lower</u> than B at current position | Δh | A <u>slightly favored</u> | 15 |
| 7 | (A is <u>in-trail</u> behind another aircraft) AND (A has <u>in-trail</u> ETA with another aircraft) | $\Delta d \ \& \ \tau_{Ai}$ | A <u>marginally favored</u> | 7.5 |
| 8 | (B is <u>in-trail</u> behind another aircraft) AND (B has <u>in-trail</u> ETA with another aircraft) | $\Delta d \ \& \ \tau_{Bi}$ | A <u>marginally favored</u> | 7.5 |
| 9 | A sequenced ahead of B <u>causes less delay</u> | NDS | A <u>marginally favored</u> | 7.5 |
| 10 | B sequenced behind A is <u>out of delay</u> | κ | B <u>significantly favored</u> | -45 |
| 11 | B sequenced behind A is <u>significantly delayed</u> | ζ | B <u>favored</u> | -30 |
| 12 | B sequenced behind A is <u>delayed</u> | ζ | B <u>slightly favored</u> | -15 |
| 13 | B sequenced behind A is <u>slightly delayed</u> | ζ | B <u>marginally favored</u> | -7.5 |
| 14 | B <u>ahead</u> of A <u>along downwind</u> | NSD_{DOWN} | B <u>somewhat favored</u> | -22.5 |
| 15 | B <u>lower</u> than A at current position | Δh | B <u>slightly favored</u> | -15 |
| 16 | (B is <u>in-trail</u> behind another aircraft) AND (B has <u>in-trail</u> ETA with another aircraft) | $\Delta d \ \& \ \tau_{Ai}$ | B <u>marginally favored</u> | -7.5 |
| 17 | (A is <u>in-trail</u> behind another aircraft) AND (A has <u>in-trail</u> ETA with another aircraft) | $\Delta d \ \& \ \tau_{Bi}$ | B <u>marginally favored</u> | -7.5 |
| 18 | B sequenced ahead of A <u>causes less delay</u> | NDS | B <u>marginally favored</u> | -7.5 |

In Table 2-4, ETA is the Estimated Time of Arrival of an aircraft at the runway edge.

For ordering aircraft which have a GENERAL-Type Spatial Constraint, each aircraft in the initial Spatial Constraint is compared to every other in that Spatial Constraint. When merging aircraft of a GENERAL-Type Spatial Constraint, the lead aircraft from each of the separate Spatial Constraints are compared, while maintaining the relative sequence established by the initial Spatial Constraint ordering Procedure.

The repetition in Table 2-1 of Propositions 2 and 4 with different Consequents, and Propositions 9 and 11 with different Consequents is based on the fact that the Propositions for the ordering and merging Procedure of a GENERAL Type Spatial Constraint are conveniently implemented as those in Table 2-2. However, NSD_{FCTS} is interpreted as NSD_{TRK} in Table 2-1.

Since the aircraft pass from separate flight path segments onto a common segment at different times, an important common reference point is the FCTS for the Merging Procedure of a GENERAL-type Spatial Constraint. It is defined to be the earliest time both aircraft reach the same flight path segment based on their predicted position, or equivalently, the first time they both belong to the same Spatial Constraint.

Each of the four Spatial Constraint scheduling decisions is based on set of fuzzy logic Propositions. The inputs to these fuzzy logic Propositions are relative positions (Δd), relative ground speed (ΔV_G), relative altitude (Δh), ETA, or derived variables based on the previous four variables, for two aircraft at a time. The derived variables are the NSD, the Controllability (κ), the Excess Delay (ζ), NDS, and the Relative ETA Magnitude (τ). The equations for these derived variables are presented in a later section.

These inputs are translated, via Membership Functions, into a Membership value which varies between 0 and 1. Zero indicates no membership, one indicates complete membership, while intermediate values indicate the degree of membership for the two aircraft for that particular Proposition.

The Membership value becomes the independent variable used by the assigned Consequent function to define an Output and an Output Weight. The Output values, which are summarized in Table 2-5, have been selected arbitrarily.

The Output Weight, which varies between 0 and 5, is determined by the Membership value. By summing the weighted Outputs, the so-called Firing Strengths, from all the applicable Propositions for a particular Spatial Constraint, a Procedure Decision Score, or Firing Strength, is obtained. The polarity of that Score determines, in part, if the previous Decision (order of the two aircraft) will be maintained (+) or reversed (–). In addition, hysteresis is built into this Decision logic such that a reversal of a previous Decision will not be implemented until the new Procedure Normalized Firing Strength has reached a value of at least ± 7.5 .

Table 2-5. Proposition Output Values

| Consequent Function | Output |
|-------------------------|--------|
| 'Marginally favored' | 7.5 |
| 'Slightly favored' | 15 |
| 'Somewhat favored' | 22.5 |
| 'Favored' | 30 |
| 'Quite favored' | 37.5 |
| 'Significantly favored' | 45 |

2.3 Proposition Inputs

In this section, the Membership Function input variables will be defined and the equations used to compute these variables will be presented. It will be shown that while the FAST Trajectory Synthesizer (TS) provides the position, velocity, ETA, and the relative position and ground speed at FCTS, the FAST SL uses some of these as input variables directly and others are used to compute secondary input variables.

It is convenient to combine a number of the individual Propositions and their inputs into Proposition pairs, if the Membership Functions are defined in terms of criteria which depend on relative ground speed sensitive inputs. This facilitates the Proposition error analysis, presented in Chapter 3. Specifically this combined formulation may show that the Propositions may be erroneously satisfied due to ground speed errors. The remaining Propositions, which depend on the current order of the two aircraft (e.g.: aircraft A is ahead of aircraft B, or aircraft A is below aircraft B) define a unique set of conditions which are insensitive to these ground speed errors.

A number of input variables involve the NSD between two aircraft (A and B). These are the NSD_{FCTS} , the current NSD (NSD_{TRK}), and the NSD along the downwind segment (NSD_{DOWN}). These use as inputs the relative current or future path distance of two aircraft with the position referenced to a common arbitrary origin. This origin is selected to be closer to the runway than the Metering Fix and positive in the direction of the Metering Fix. As a result, the lead aircraft will have a shorter path distance than the trailing aircraft relative to this arbitrary reference point. These inputs are defined as follows for the general case of A ahead of B or vice versa:

2.3.1 Normalized Separation Distance at FCTS

The primary criteria for ordering two aircraft currently on different Spatial Constraints is the Normalized Separation Distance (NSD) at their first common time step (FCTS) on

the same Spatial Constraint. This variable determines which aircraft is ahead at FCTS and favors this sequence for the two aircraft.

$$NSD_{FCTS}(t_k) \equiv \begin{cases} \frac{(d_{B,FCTS}(t_k) - d_{A,FCTS}(t_k))}{\Delta d_{AB}}, & \text{if, } d_{B,FCTS}(t_k) > d_{A,FCTS}(t_k) \\ \frac{(d_{B,FCTS}(t_k) - d_{A,FCTS}(t_k))}{\Delta d_{BA}}, & \text{if, } d_{B,FCTS}(t_k) \leq d_{A,FCTS}(t_k) \end{cases}$$

where, d_{FCTS} = predicted path distance at FCTS relative to arbitrary reference point
 d_{AB} = minimum required separation distance in the TRACON given an aircraft order: A followed by B (See Table 2-6).

Table 2-6. Aircraft Minimum Required Separation Distance, Δd_{AB} (nmi.)

| Leading Aircraft, A | Trailing Aircraft, B | | | |
|---------------------|----------------------|-------|-------|------|
| | Small | Large | Heavy | B757 |
| Small | 3 | 3 | 3 | 3 |
| Large | 4 | 3 | 3 | 3 |
| Heavy | 6 | 5 | 4 | 5 |
| B757 | 4 | 4 | 4 | 4 |

2.3.2 Normalized Separation Distance at Current Position (NSD_{TRK})

The Normalized Separation Distance at the current position determines the current order of two aircraft relative to the runway threshold. Positive values indicate that aircraft A is currently ahead of B while negative values indicate the reverse order.

$$NSD_{TRK}(t_k) \equiv \begin{cases} \frac{(d_{B,TRK}(t_k) - d_{A,TRK}(t_k))}{\Delta d_{AB}}, & \text{if, } d_{B,TRK}(t_k) > d_{A,TRK}(t_k) \\ \frac{(d_{B,TRK}(t_k) - d_{A,TRK}(t_k))}{\Delta d_{BA}}, & \text{if, } d_{B,TRK}(t_k) \leq d_{A,TRK}(t_k) \end{cases}$$

where, d_{TRK} = aircraft current path distance relative to arbitrary reference point

2.3.3 Normalized Separation Distance along Downwind Segment

The Normalized Separation Distance along the Downwind Segment is used to sequence two aircraft if their final approach course has been extended with a base extension. It is used to favor a sequence which considers the expected order of the two aircraft on the Downwind Segment. A positive value indicates aircraft A is further along the Downwind Segment than aircraft B while a negative value indicates the reverse.

$$NSD_{DOWN}(t_k) = \begin{cases} \frac{(d_{B,DOWN}(t_k) - d_{A,DOWN}(t_k))}{\Delta d_{AB}}, & \text{if, } d_{B,DOWN}(t_k) > d_{A,DOWN}(t_k) \\ \frac{(d_{B,DOWN}(t_k) - d_{A,DOWN}(t_k))}{\Delta d_{BA}}, & \text{if, } d_{B,DOWN}(t_k) \leq d_{A,DOWN}(t_k) \end{cases}$$

where, d_{DOWN} = path distance to Downwind Segment relative to arbitrary reference

2.3.4 Horizontal Separation

Two additional distance-based Proposition input variables are the Horizontal Separation and the Altitude Separation. The Horizontal Separation determines the line-of-sight horizontal separation between the two aircraft. This will be different from the path separation distance if the two aircraft are not on the same flight path segment.

$$\Delta d = \sqrt{(x_B - x_A)^2 + (y_B - y_A)^2}$$

where, x_i, y_i = radar coordinate positions of aircraft i

This variable is also used together with the relative ground speed of two aircraft to determine whether an overtake is desirable. It is also used together with the relative ETA magnitude to determine whether two aircraft are in-trail of each other -- one behind the other.

2.3.5 Altitude Separation

The Altitude Separation between two aircraft, A and B, is used to establish which aircraft is currently at a lower altitude. This variable is used for ordering aircraft which are stacked on top of each other.

$$\Delta h = (h_{B,TRK} - h_{A,TRK})$$

where, h_{TRK} = current altitude of the aircraft.

A positive value indicates that aircraft A is lower than aircraft B while a negative value indicates the reverse.

2.3.6 Relative Ground Speed

The Relative Ground Speed between two aircraft, either at their current position or at their FCTS, determines which aircraft is faster. This variable is used to identify overtakes at the current position or at their FCTS, and this is used to determine whether these overtakes are acceptable.

The Relative Ground Speed input variable is defined for two aircraft as:

$$\Delta V_G = (V_{G,A} - V_{G,B})$$

where, V_G = current ground speed of aircraft.

If it is positive, then aircraft A is faster than aircraft B while if it is negative, the reverse is true.

2.3.7 Relative ETA Magnitude

Four of the input variables are based on the flight time of the aircraft. As a result, they indirectly involve the current measured position and ground speed of the aircraft. The four input variables are the current Relative ETA Magnitude (τ_{AB}), the Controllability (κ), the Excess Delay (ζ), and the NDS.

The Relative ETA Magnitude of two aircraft is used as an indicator of whether two aircraft should be considered to be in-trail -- one behind the other. This input variable is used together with the Horizontal Separation to establish that the two aircraft are indeed in-trail of each other.

$$\tau_{AB} = |TA_{B,Early} - TA_{A,Early}|$$

where, $TA_{A,Early}, TA_{B,Early}$ = earliest times of arrival for aircraft A and B

The earliest time of arrival, TA_{EARLY} , is obtained by determining the time it takes an aircraft to reach the runway threshold when it takes the shortest flight path and the latest speed reductions.

2.3.8 Controllability

The Controllability determines if an aircraft is unable to absorb enough delay to achieve a desired sequence between two aircraft. A positive value indicates that a desired sequence of aircraft A ahead of aircraft B allows aircraft B to have excess delay. A

negative value, in turn, indicates that this sequence will require more delay of aircraft B than it has available, based on the aircraft's ETA range. The time of arrival range is determined by the earliest and latest arrival trajectory for that aircraft.

$$\kappa \equiv \begin{cases} \frac{(TA_{A,Late} - STA_{A,BA})}{\Delta t_{BA}}, & \text{if A is sequenced behind B} \\ -\frac{(TA_{B,Late} - STA_{B,AB})}{\Delta t_{AB}}, & \text{if B is sequenced behind A} \end{cases}$$

where, $STA_{A,AB}$ = scheduled time of arrival of aircraft A, given a sequence of AB
 $STA_{B,AB} = \max\{ STA_{A,AB} + \Delta t_{AB}, TA_{B,Early} \}$, with $STA_{1stAircraft} = TA_{1st,Early}$
 TA_{Late} = latest time of arrival for that aircraft
 Δt_{jk} = minimum required separation time between aircraft given a sequence of aircraft j followed by aircraft k

The latest time of arrival, TA_{Late} , is obtained by determining the time required for an aircraft to reach the runway threshold when it takes the longest flight path, using any flight path extensions, and the earliest speed reductions.

The scheduled time of arrival, $STA_{B,AB}$, is obtained from the earliest times of arrival of aircraft B and the scheduled time of arrival of the aircraft A. If aircraft A is the first aircraft of that particular path segment, then its earliest time of arrival defines its scheduled time of arrival. For all other aircraft, the scheduled time of arrival is determined by the maximum of the aircraft's earliest time of arrival and the sum of the scheduled time of arrival of the aircraft in front of it plus their minimum required time separation.

The minimum required separation time between two aircraft is computed by dividing the required separation distance of Table 2-6 by an aircraft speed of 170 knots for all aircraft. This produces the minimum required time separation shown in Table 2-7.

Table 2-7. Aircraft Minimum Relative Time Spacing, Δt_{AB} (min)*

| Leading Aircraft, A | Trailing Aircraft, B | | | |
|---------------------|----------------------|-------|-------|------|
| | Small | Large | Heavy | B757 |
| Small | 1.0 | 1.0 | 1.0 | 1.0 |
| Large | 1.4 | 1.0 | 1.0 | 1.0 |
| Heavy | 2.1 | 1.8 | 1.4 | 1.8 |
| B757 | 1.4 | 1.4 | 1.4 | 1.4 |

* Assumes a ground speed of 170 knots for all aircraft

2.3.9 Excess Delay

The Excess Delay, expressed as a percentage of the available delay, is used to favor sequences which distribute the required delays equally among all aircraft. It determines the amount of delay not used to achieve the required aircraft sequence and to maintain the required separation at the runway threshold. Values near 100% indicate that little additional flight time is required to achieve the required sequence. Values near zero, in turn, indicate that a large amount of the aircraft's available delay is used to achieve the required sequence.

$$\zeta = 100 \cdot \begin{cases} \left(\frac{TA_{A,Late} - STA_{A,BA}}{TA_{A,Late} - TA_{A,Early}} \right), & \text{if A is sequenced behind B} \\ - \left(\frac{TA_{B,Late} - STA_{B,AB}}{TA_{B,Late} - TA_{B,Early}} \right), & \text{if B is sequenced behind A} \end{cases}$$

2.3.10 Normalized Delay Savings (NDS)

The Normalized Separation Distance indicates whether the current relative sequence of two aircraft or the reversed order decreases their overall flight time. A positive NDS indicates that the order of aircraft A ahead of B causes less delay than the reverse order.

$$NDS = \frac{(STA_{B,BA} - TA_{B,Early}) + (STA_{A,BA} - TA_{A,Early})}{\Delta t_{BA}} - \frac{(STA_{A,AB} - TA_{A,Early}) + (STA_{B,AB} - TA_{B,Early})}{\Delta t_{AB}}$$

2.4 Proposition Results

It is of interest in Chapter 4 to know the nominal equations for the Proposition Membership Function, Output, Weight, and the Weighted Output, or the so-called 'Firing Strength.' These variables are expressed in terms of the Proposition input functions. The equations for the Membership Function, Output, Weight and the Firing Strength will be presented in this section for the Propositions whose input values depend on the Relative Ground Speed directly or indirectly.

The individual Propositions of Tables 2-1 through 2-4 are presented separately for aircraft A versus B and then again for aircraft B versus A. It is desirable to combine the Membership Functions for two Propositions if the condition for first or second Proposition depends on a relative ground speed dependent variable. This makes it possible to determine if errors in these relative ground speed variables lead to a reversal of the conditions favoring one or the other of the two Propositions.

To be more specific, the Propositions which have conditions which are sensitive to relative ground speed errors are the Relative Ground Speed itself and the NSD_{FCTS} . The latter satisfies this criteria because the NSD_{FCTS} depends on the predicted order of the two aircraft under consideration, rather than on their current order.

This criteria is not satisfied by the Controllability, Excess Delay, and Normalized Delay Savings, since the condition used for their associated Propositions is whether aircraft A is currently ahead (or sequenced ahead) of aircraft B or whether the reverse is true. For the Relative ETA Magnitude, there is only one Membership-Consequent function pair which is evaluated jointly with the Horizontal Separation of the two aircraft.

2.4.1 Relative Ground Speed

In this section the equations describing Proposition Membership Function, Output, Weight, and Firing Strength as a function of the nominal Relative Ground Speed are presented. The Relative Ground Speed Membership Function: 'Is faster' and the corresponding Consequent Function: 'Is marginally favored' is illustrated in Figure 2-1.

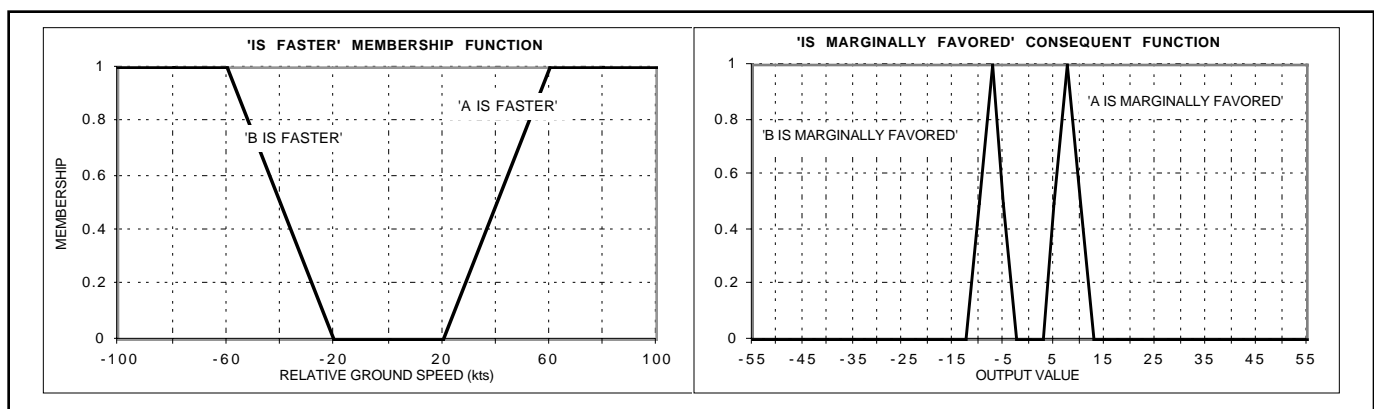


Figure 2-1. Relative Ground Speed Membership and Consequent Functions

The left-hand Membership Function ('B is faster than A') is used with the left-hand Consequent Function ('B marginally favored over A'). While the right-hand Membership Function ('A is faster than B') is used with the right-hand Consequent Function ('A marginally favored over B'). Note that this Proposition has only three possible Outputs: zero, -7.5, and +7.5. The Weight associated with this Output is determined by the fraction of the triangular area beneath the Membership value for a specific Relative Ground Speed.

For a Relative Ground Speed, v , the Membership Function 'Is faster', $M(v)$, is given by:

$$M(v) = \begin{cases} 1, & \text{for, } v < -60 \text{ kts} \\ -\frac{(20 + v)}{40}, & \text{for, } -60 \text{ kts} \leq v < -20 \text{ kts} \\ 0, & \text{for, } -20 \text{ kts} \leq v < 20 \text{ kts} \\ \frac{(v - 20)}{40}, & \text{for, } 20 \text{ kts} \leq v < 60 \text{ kts} \\ 1, & \text{for, } 60 \text{ kts} \leq v \end{cases}$$

The Consequent Function: 'Is marginally favored,' has an Output, $O(v)$, which is described by:

$$O(v) = \begin{cases} -7.5, & \text{for, } v < -20 \text{ kts} \\ 0, & \text{for, } -20 \text{ kts} \leq v < 20 \text{ kts} \\ 7.5, & \text{for, } 20 \text{ kts} \leq v \end{cases}$$

The Weight, $W(v)$, which is associated with the Output, $O(v)$, for the Membership Function, $M(v)$, is given by:

$$W(M) = \begin{cases} 0, & \text{for } M = 0 \\ 5M(2 - M), & \text{for, } 0 < M < 1 \\ 5, & \text{for } M = 1 \end{cases}$$

Substituting for $M(v)$ into this last expression, the Weight as a function of the Relative Ground Speed, v , is obtained:

$$W(v) = \begin{cases} 5, & \text{for, } v < -60 \text{ kts} \\ -\frac{(v+20)(v+100)}{320}, & \text{for, } -60 \text{ kts} \leq v < -20 \text{ kts} \\ 0, & \text{for, } -20 \text{ kts} \leq v < 20 \text{ kts} \\ -\frac{(v-20)(v-100)}{320}, & \text{for, } 20 \text{ kts} \leq v < 60 \text{ kts} \\ 5, & \text{for, } 60 \text{ kts} \leq v \end{cases}$$

Using the expression for the Output, $O(v)$, and the Weight, $W(v)$, the Firing Strength, $S(v)$, can be computed as the Weighted Output:

$$S(v) = W(v)O(v)$$

or,

$$S(v) = \begin{cases} -37.5, & \text{for, } v < -60 \text{ kts} \\ \frac{3(v+20)(v+100)}{128}, & \text{for, } -60 \text{ kts} \leq v < -20 \text{ kts} \\ 0, & \text{for, } -20 \text{ kts} \leq v < 20 \text{ kts} \\ -\frac{3(v-20)(v-100)}{128}, & \text{for, } 20 \text{ kts} \leq v < 60 \text{ kts} \\ 37.5, & \text{for, } 60 \text{ kts} \leq v \end{cases}$$

These equations now make it possible to compute the Membership Function, Output, Weight, and Firing Strength for any specific value of the Relative Ground Speed which is input.

2.4.2 Normalized Separation Distance at FCTS

The NSD_{FCTS} Propositions are illustrated in Figure 2-2. As shown in Section 2.3, the calculation of the NSD_{FCTS} may differ depending on the order of the aircraft, whether AB or BA. This arises from the use of a normalization factor which depends on what type of aircraft is ahead of what other type of aircraft.

These three Membership and Consequent Function pairs are seen to have a similar structure to that of the Relative Ground Speed Membership and Consequent Function pair. Hence, it is worthwhile to define the general equations for the Membership Function, Output, Weigh, and Firing Strength for any general input.

If x is a general input to a trapezoidal Membership Function, $M(x)$:

$$M(x) = \begin{cases} 1, & \text{for, } x < b_L \\ \frac{(x - a_L)}{(b_L - a_L)}, & \text{for, } b_L \leq x < a_L \\ 0, & \text{for, } a_L \leq x < a_R \\ \frac{(x - a_R)}{(b_R - a_R)}, & \text{for, } a_R \leq x < b_R \\ 1, & \text{for, } b_R \leq x \end{cases}$$

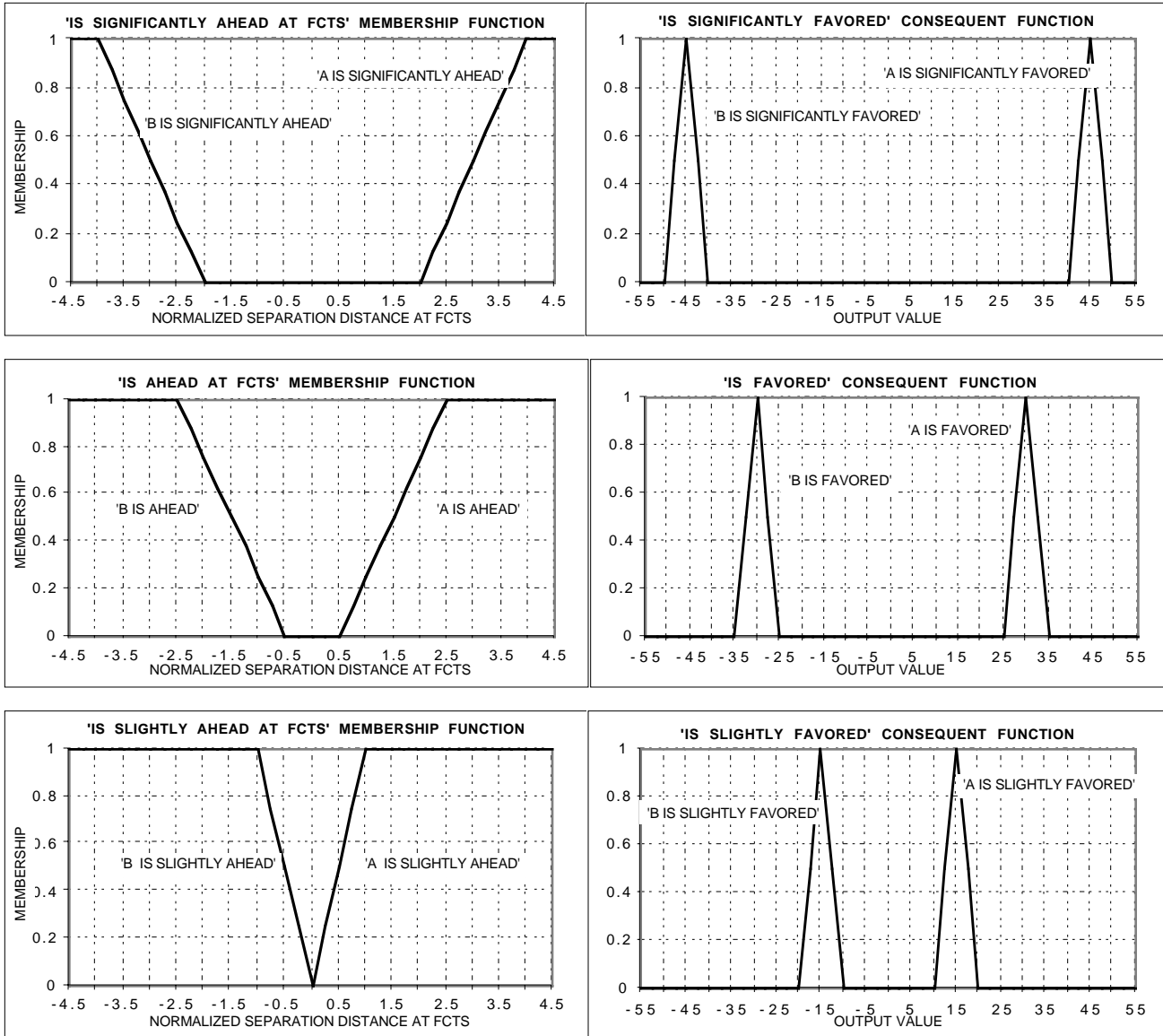


Figure 2-2. NSD_{FCTS} Propositions

where a_L, a_R = left and right-hand input limits of trapezoidal dead-band ($M = 0$)
 b_L, b_R = left and right-hand input limits where trapezoid reaches its maximum value ($M = 1$)

In this definition of the trapezoidal Membership Function, the left-hand limits are associated with a negative slope, while those for the positive limits are associated with a positive slope.

The general Output Function $O(x)$ based on the input x is:

$$O(x) = \begin{cases} -c, & \text{for, } x < a_L \\ 0, & \text{for, } a_L \leq x < a_R \\ c, & \text{for, } a_R \leq x \end{cases}$$

where c = Consequent Function discrete Output Value (Table 2-5)

The general Output Weight, $W(x)$, is given by:

$$W(x) = \begin{cases} 5, & \text{for, } x < b_L \\ -\frac{5(x - a_L)(x + a_L - 2b_L)}{(b_L - a_L)^2}, & \text{for, } b_L \leq x < a_L \\ 0, & \text{for, } a_L \leq x < a_R \\ -\frac{5(x - a_R)(x + a_R - 2b_R)}{(b_R - a_R)^2}, & \text{for, } a_R \leq x < b_R \\ 5, & \text{for, } b_R \leq x \end{cases}$$

Then the general Firing Strength, $S(x)$, can be computed as the Weighted Output:

$$S(x) = W(x)O(x)$$

or,

$$S(x) = \begin{cases} -5c, & \text{for, } x < b_L \\ \frac{5c(x - a_L)(x + a_L - 2b_L)}{(b_L - a_L)^2}, & \text{for, } b_L \leq x < a_L \\ 0, & \text{for, } a_L \leq x < a_R \\ -\frac{5c(x - a_R)(x + a_R - 2b_R)}{(b_R - a_R)^2}, & \text{for, } a_R \leq x < b_R \\ 5c, & \text{for, } b_R \leq x \end{cases}$$

Hence for the three NSD_{FCTS} Propositions of Figure 2-2, the parameters are summarized in Table 2-8.

Table 2-8 NSD_{FCTS} Proposition Parameters

| Membership Function | b_L | a_L | a_R | b_R | Consequent Function | c |
|-------------------------------|-------|-------|-------|-------|-------------------------|----|
| 'Significantly Ahead at FCTS' | -4 | -2 | 2 | 4 | 'Significantly Favored' | 45 |
| 'Ahead at FCTS' | -2.5 | -0.5 | 0.5 | 2.5 | 'Favored' | 30 |
| 'Slightly Ahead at FCTS' | -1 | 0 | 0 | 1 | 'Slightly Favored' | 15 |

2.4.3 Relative ETA Magnitude

The relative ETAs between two neighboring aircraft which are in-trail on the same flight path segment, are described by the Membership and Consequent Functions shown in Figure 2-3.

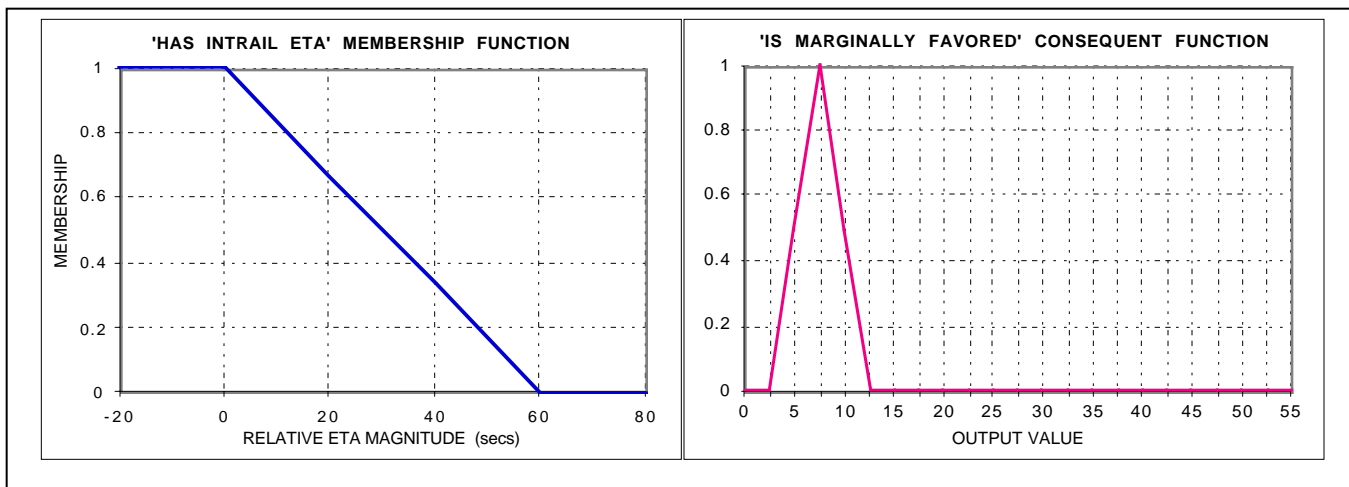


Figure 2-3 Relative ETA Proposition

The general equations for the Membership Function, Output, Weight, and Firing strength can be used with the parameters of Table 2-9.

Table 2-9. Relative ETA Magnitude Proposition Parameters

| Membership Function | b_L | a_L | a_R | b_R | Consequent Function | c |
|---------------------|-------|-------|-------|-------|----------------------|-----|
| 'Has In-Trail ETA' | 0 | 60 | 0 | 0 | 'Marginally Favored' | 7.5 |

2.4.4 Excess Delay

The three Excess Delay Propositions for aircraft B sequenced behind A are illustrated in Figure 2-4. The three corresponding Propositions for aircraft A sequenced behind B are illustrated in Figure 2-5.

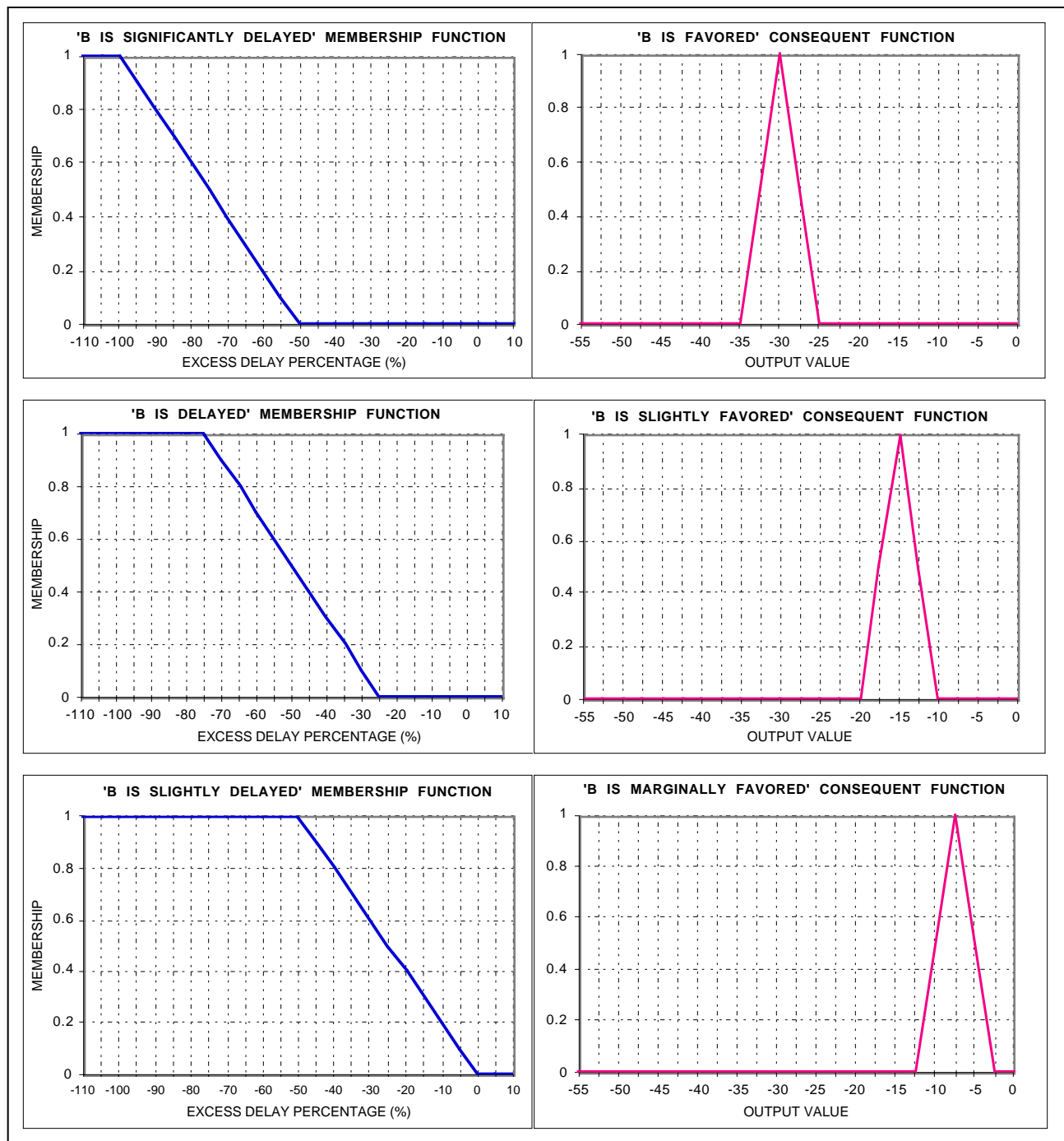


Figure 2-4. 'B Sequenced behind A' Excess Delay Propositions

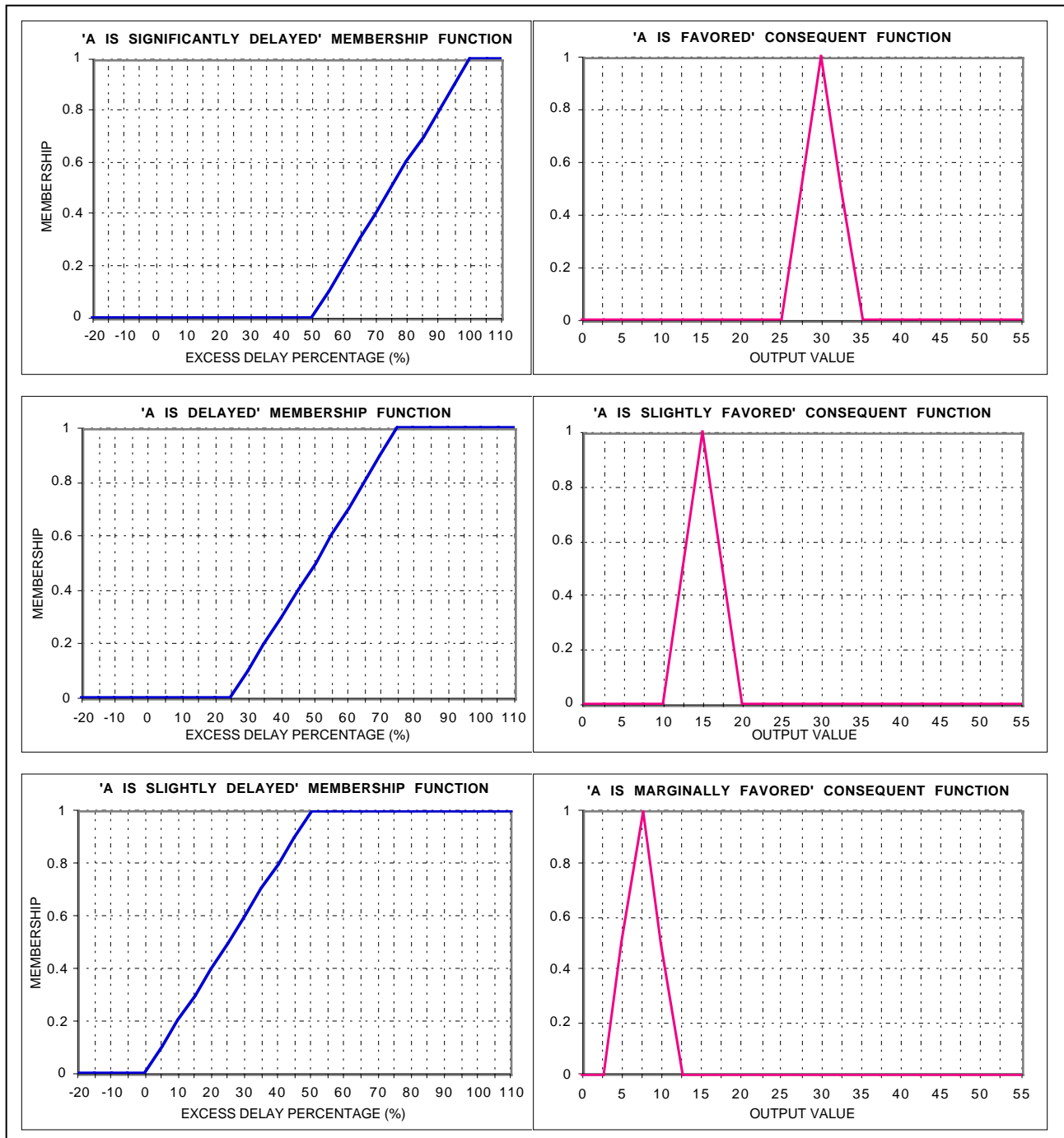


Figure 2-5. 'A Sequenced behind B' Excess Delay Propositions

The Excess Delay Membership Function, Output, Weight, and Firing Strength can be computed using the general formulas derived in Section 2.4.3. The values of the Proposition parameters are summarized in Tables 2-10 and 2-11, respectively for aircraft A sequenced behind B and for aircraft B sequenced behind A.

Table 2-10 Excess Delay Proposition Parameters (Aircraft A Sequenced Behind B)

| Membership Function | b_L | a_L | a_R | b_R | Consequent Function | c |
|------------------------------|-------|-------|-------|-------|---------------------------|-----|
| 'A is Significantly Delayed' | 0 | 0 | 50 | 100 | 'A is Favored' | 30 |
| 'A is Delayed' | 0 | 0 | 25 | 75 | 'A is Slightly Favored' | 15 |
| 'A is Slightly Delayed' | 0 | 0 | 0 | 50 | 'A is Marginally Favored' | 7.5 |

Table 2-11 Excess Delay Proposition Parameters (Aircraft B Sequenced Behind A)

| Membership Function | b_L | a_L | a_R | b_R | Consequent Function | c |
|------------------------------|-------|-------|-------|-------|---------------------------|------|
| 'B is Significantly Delayed' | -100 | -50 | 0 | 0 | 'B is Favored' | -30 |
| 'B is Delayed' | -75 | -25 | 0 | 0 | 'B is Slightly Favored' | -15 |
| 'B is Slightly Delayed' | -50 | 0 | 0 | 0 | 'B is Marginally Favored' | -7.5 |

2.4.5 Normalized Delay Savings (NDS)

The Normalized Delay Savings Proposition includes only one membership-Consequent function pair. It is illustrated in Figure 2-6.

The Proposition parameters which are required to compute the Proposition Membership Function, Output, Weight, and Firing Strength are summarized in Table 2-12.

Table 2-12 NDS Proposition Parameters

| Membership Function | b_L | a_L | a_R | b_R | Consequent Function | c |
|----------------------------------|-------|-------|-------|-------|---------------------------|------|
| 'A Ahead of B Causes Less Delay' | 0 | 0 | 0 | 2 | 'A is Marginally Favored' | 7.5 |
| 'B Ahead of A Causes Less Delay' | -2 | 0 | 0 | 0 | 'B is Marginally Favored' | -7.5 |

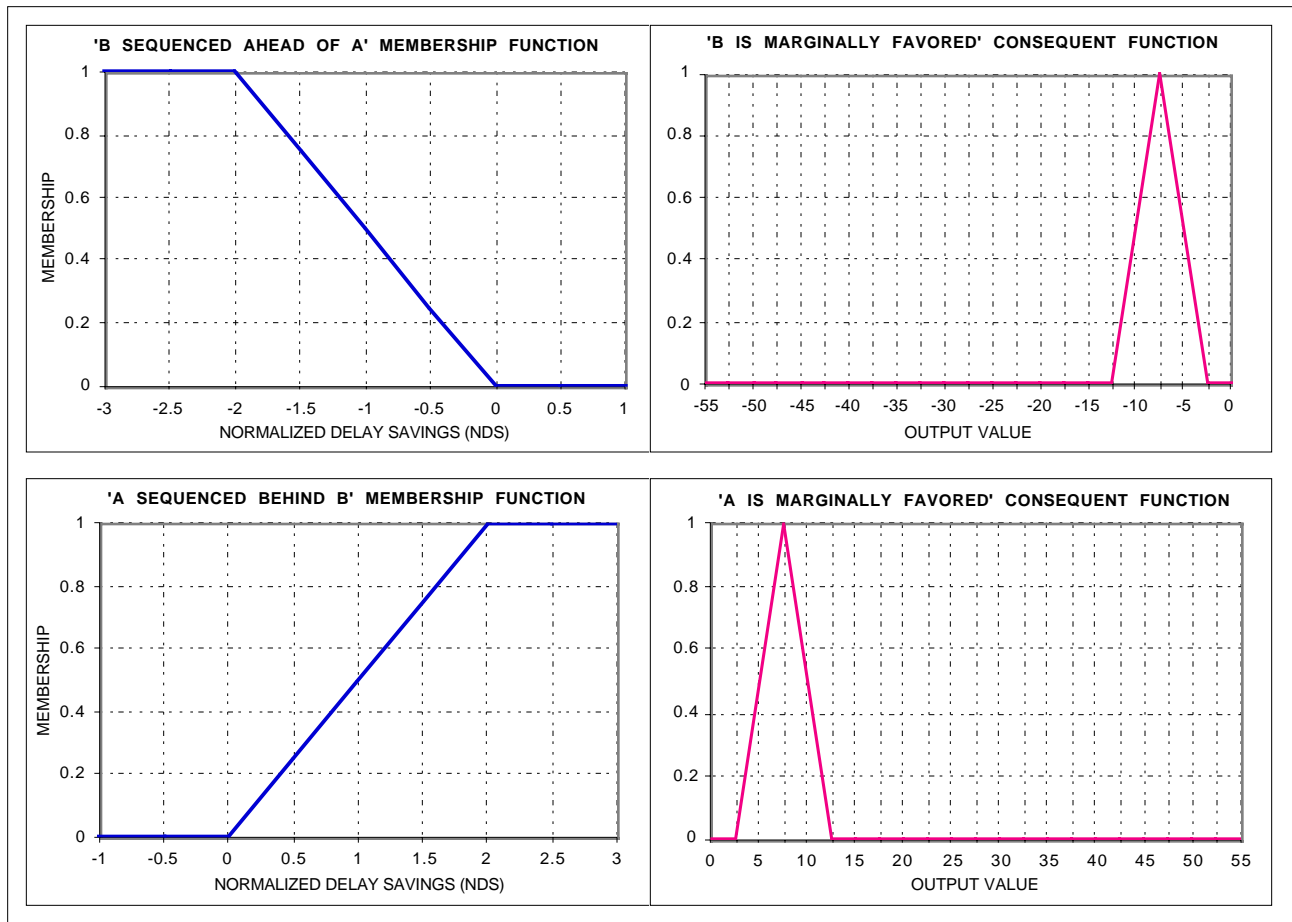


Figure 2-6. Normalized Delay Savings Propositions

2.4.6 Controllability (κ)

There are four Controllability Propositions which are illustrated in Figure 2-7 for aircraft A sequenced behind B and in Figure 2-8 for aircraft B sequenced behind A. To obtain the Controllability Proposition Membership Function, Output, Weight, and Firing Strength equations, it is convenient to use the parameters of Table 2-13.

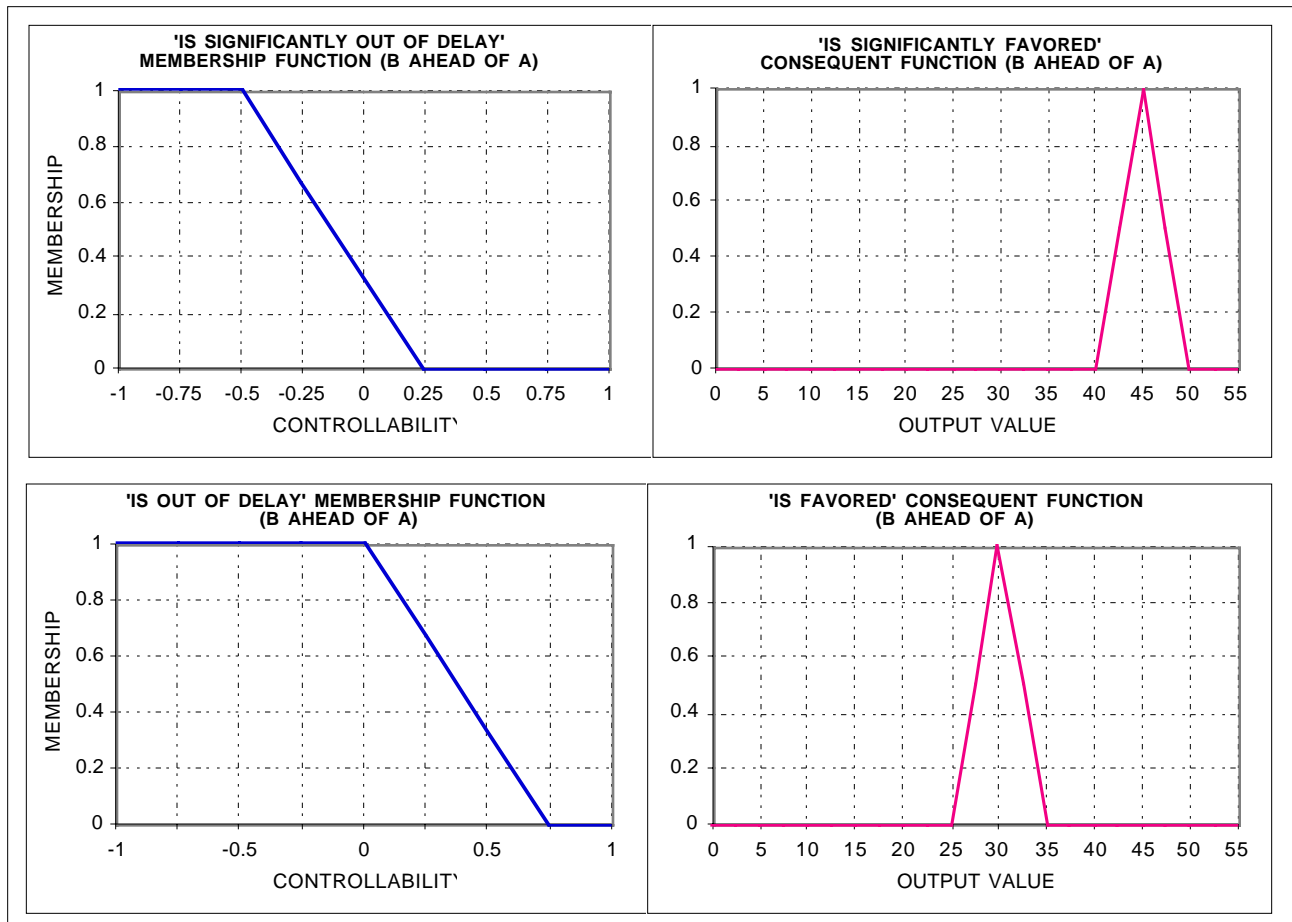


Figure 2-7 Controllability Propositions (A Sequenced Behind B)

Table 2-13. Controllability Proposition Parameters

| Membership Function | b_L | a_L | a_R | b_R | Consequent Function | c |
|---|-------|-------|-------|-------|---------------------------|-----|
| 'Significantly out of Delay' (Aircraft A behind B) | -0.5 | 0.25 | 0 | 0 | 'A Significantly Favored' | 45 |
| 'Significantly out of Delay' (Aircraft B behind A) | 0 | 0 | -0.25 | 0.5 | 'B Significantly Favored' | -45 |
| 'Out of Delay' (Aircraft A behind B) | 0 | 0.75 | 0 | 0 | 'A Favored' | 30 |
| 'Out of Delay' (Aircraft B behind A) | 0 | 0 | -0.75 | 0 | 'B Favored' | -30 |

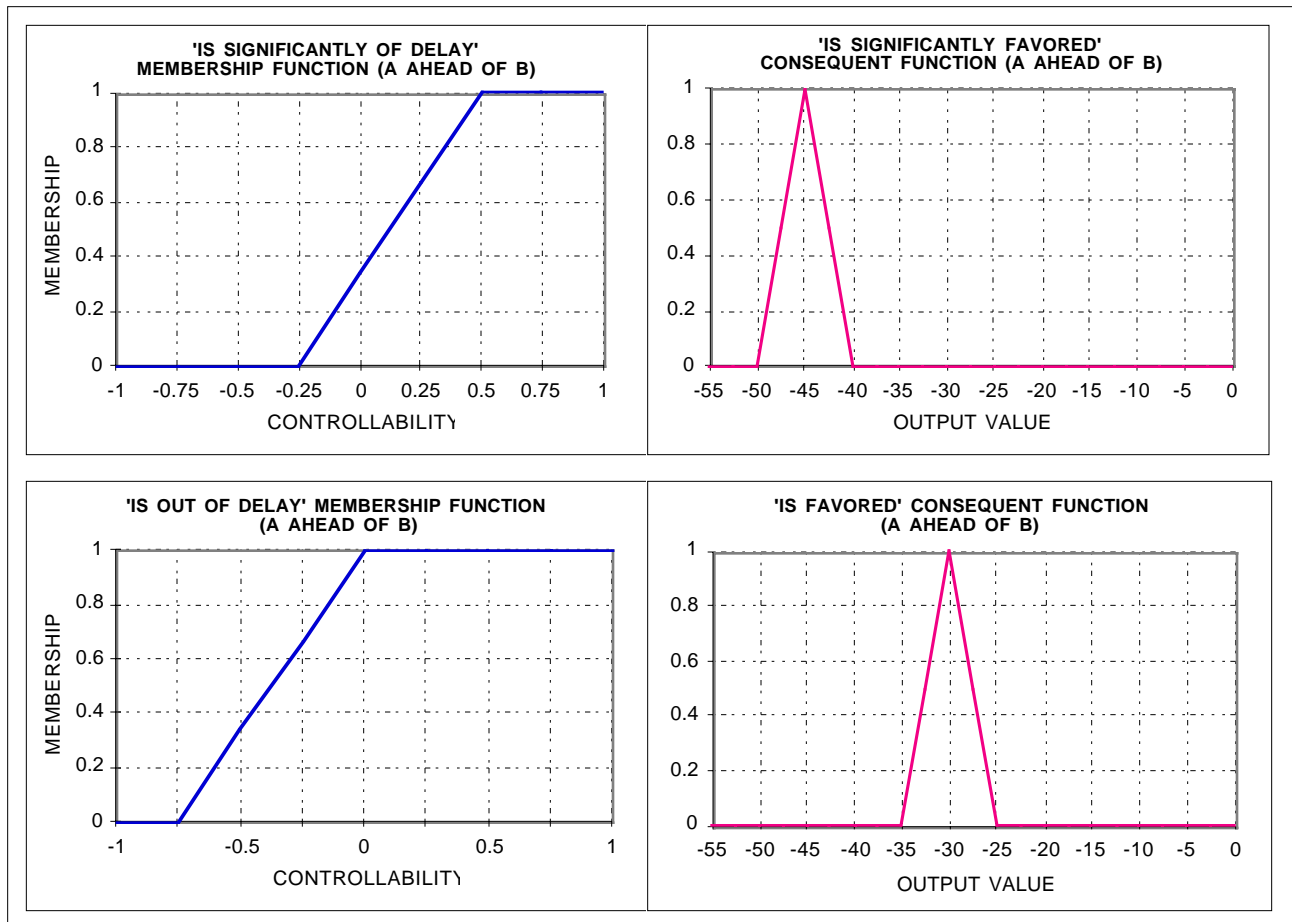


Figure 2-8 Controllability Propositions (Aircraft B Sequenced Behind A)

3.0 INPUT VARIABLE ERROR ANALYSIS

3.1 Introduction

In this Chapter, the error equations for the input variables required by FAST SL are derived. The focus is on the six input variables which depend on the Relative Ground Speed between two aircraft, whether this dependence is direct or indirect. The remaining variables are based on current position measurements. As a result these input variables are much more accurate relative to the ground speed dependent input variables. The error equations for the Relative ETA Magnitude and the Normalized Separation Distance at FCTS were previously derived in (Mueller, 1998) . As a result, these derivations are not be repeated here.

The dependence of each of the FAST SL input variables on the Horizontal Separation (Δd), Relative Altitude (Δh), and Relative Ground Speed (ΔV_G) between two aircraft is summarized in Table 3-1. It can be seen in this table that of the ten input variables, six depend on ground speed.

Table 3-1. Input Variable Dependence on Ground Speed

| Membership Function Input Variable | | Dependence on: | | |
|---|--------------|----------------|------------|--------------|
| | | Δd | Δh | ΔV_G |
| Normalized Separation Distance at FCTS | NSD_{FCTS} | Yes | | Yes |
| Normalized Separation Distance at Current Position | NSD_{TRK} | Yes | | |
| Normalized Separation Distance along Downwind Segment | NSD_{DOWN} | Yes | | |
| Horizontal Separation | Δd | Yes | | |
| Altitude Separation | Δh | | Yes | |
| Relative Ground Speed | ΔV_G | | | Yes |
| Relative ETA Magnitude | τ_{AB} | Yes | | Yes |
| Controllability | κ | Yes | | Yes |
| Excess Delay | ξ | Yes | | Yes |
| Normalized Delay Savings | NDS | Yes | | Yes |

The dependence of each of the four Procedures on the six input variables which are directly or indirectly related to the Relative Ground Speed is summarized in Table 3-2.

This table clearly shows that the Ordering Procedure of a GENERAL-Type Spatial Constraint is probably the Procedure least affected by Relative Ground Speed errors.

Table 3.2 FAST SL Ordering/Merging Procedure Dependence on Relative Ground Speed, ΔV_G

| | | Procedure* | | | |
|---|--------------|---------------------------------|-------|-------------------------------|-------|
| Relative Ground Speed-Dependent Input Variables | | GENERAL-Type Spatial Constraint | | FINAL-Type Spatial Constraint | |
| | | Order | Merge | Order | Merge |
| 1. Normalized Separation Distance at FCTS | NSD_{FCTS} | | Yes | | |
| 2. Relative Ground Speed | ΔV_G | Yes | Yes | | |
| 3. Relative ETA Magnitude | τ_{AB} | | | | Yes |
| 4. Controllability | κ | | | Yes | Yes |
| 5. Excess Delay | ζ | | | Yes | Yes |
| 6. Normalized Delay Savings | NDS | | | Yes | Yes |

3.2 Nominal TRACON Trajectories

It is of interest to determine the input variable error histories for several standard TRACON scenarios. As a result, several nominal aircraft trajectories is generated using the FAST TS simulation developed and documented in (Mueller, 1998) and presented in Appendix A.

A convenient TRACON scenario is illustrated in Figures 3-1 and 3-2. Both consist of the southwest approach to Runway 18R at Dallas-Ft. Worth. The first figure is a radar track plot of flights arriving around 9:15 AM, 11 July 1996, while the second figure is a simplified diagram of the nominal flight path segments. It can be seen that while both jets and turboprops pass over the same Metering Fix, they use slightly different flight path segments to reach the runway.

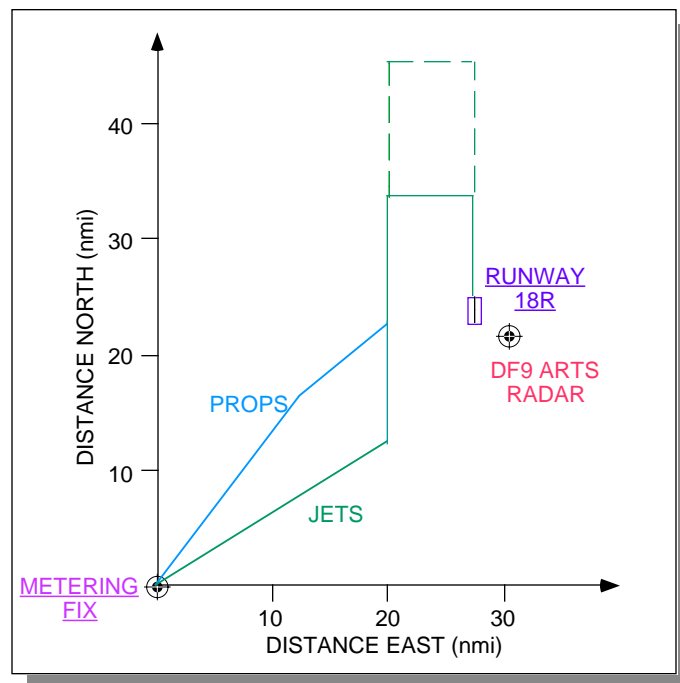
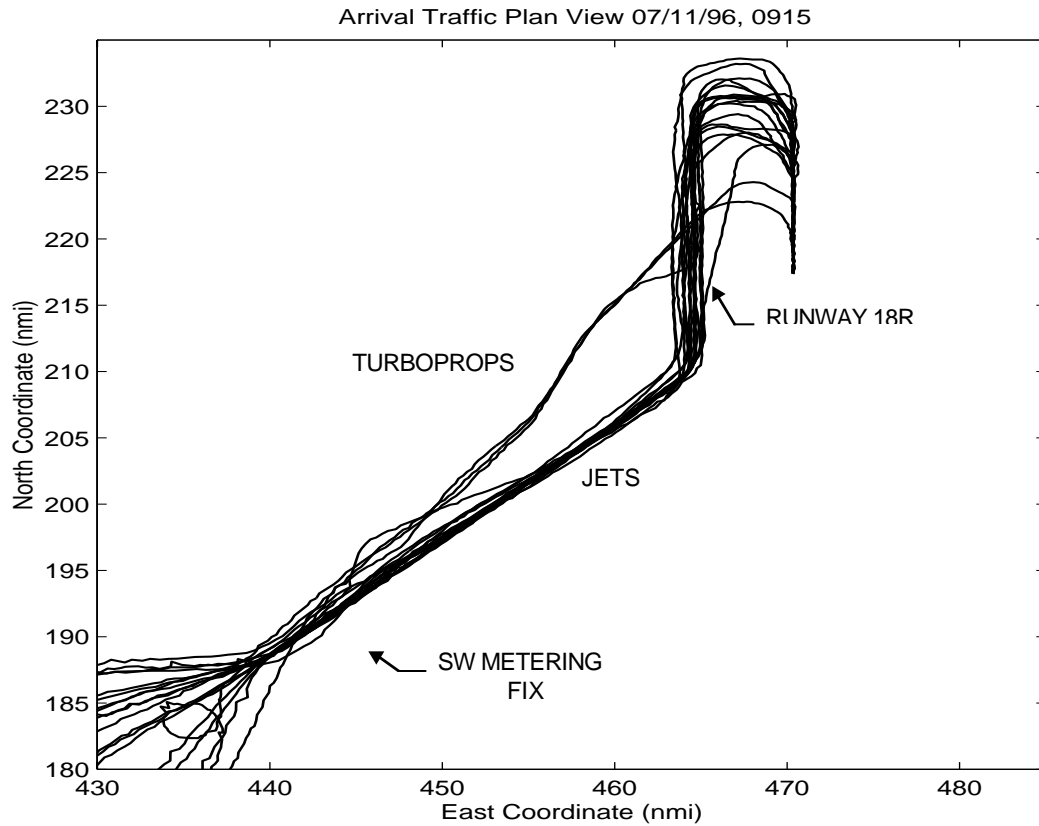


Figure 3-2. Approach to Dallas-Ft. Worth Runway 18R from SW Metering Fix

Also shown in the latter figure is a nominal base extension which would be used to obtain a late arrival trajectory. The location of the tracking radar is also shown near this runway in the latter figure.

A nominal jet and turboprop trajectory is used to provide the input variable error histories in the next section. The two trajectories are illustrated in Figure 3-3 as a function of the path distance from the runway threshold and in Figure 3-4 as a function of the time from the runway threshold.

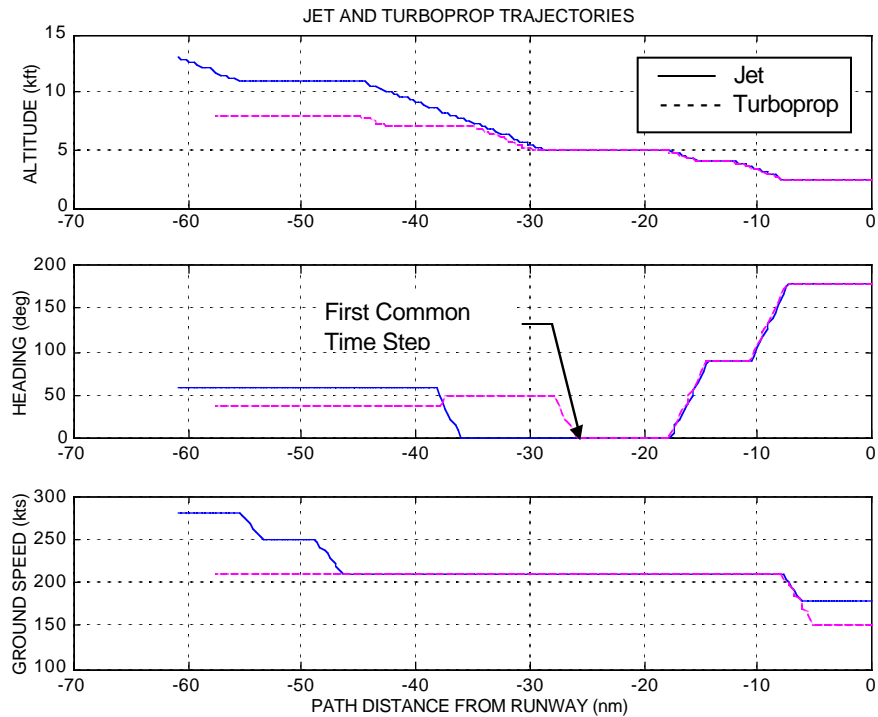


Figure 3-3. FAST TS Jet and Turboprop Distance Histories

For the input variables used by the Merging Procedure, the focus is on the time history of the two trajectories prior to and including the time of FCTS, t_{FCTS} . Since the turboprop is the last aircraft to reach the Downwind flight path segment, t_{FCTS} is determined when it reaches this segment. For this scenario, t_{FCTS} corresponds to the point when the heading of the turboprop changes to a north (0 deg.) heading. From Figures 3-3 and 3-4, it can be seen that this occurs approximately 27 nmi. or 8 min. from the runway.

To evaluate the input variables used by the Ordering Procedure, a minimum of two aircraft trajectories are required on the same flight path segment. Hence it is convenient to select two jet aircraft trajectories. To evaluate the input variables used by the Merging Procedure, requires a minimum of one aircraft each from two separate, but merging, flight path segments. Hence it is convenient to choose a jet and a turboprop trajectory. Since these conditions require a minimum of two jet and one turboprop aircraft trajectories, it is convenient to select these three such that the two jets are in near proximity to each other and that they will also be in near proximity to the turboprop at the merge point.

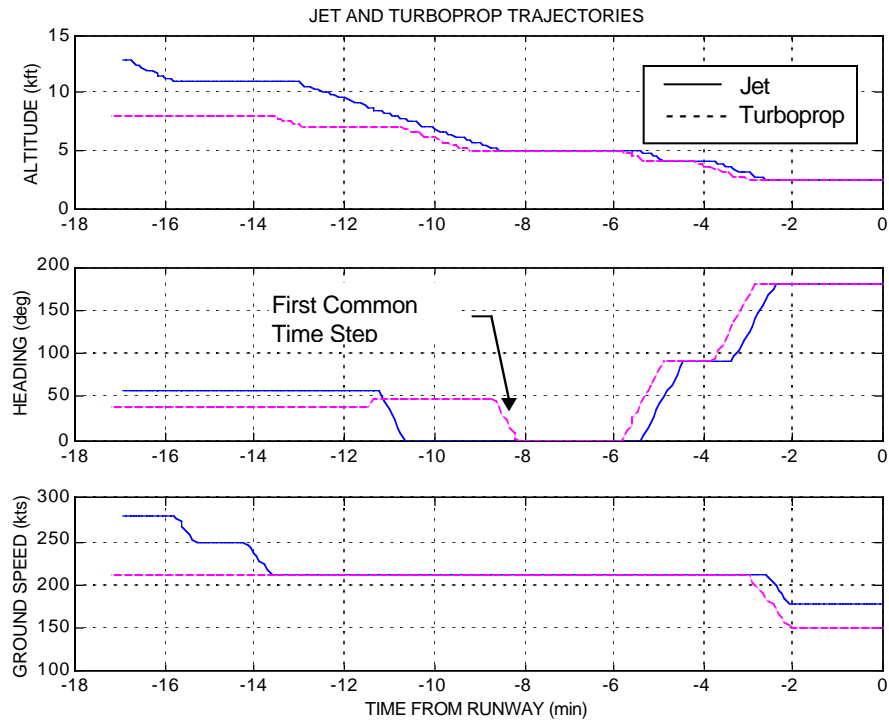


Figure 3-4. FAST TS Jet and Turboprop Time Histories

Since the input variable errors due to Relative Ground Speed errors are the principal focus of this study, it is desirable to select the relative spacing such that minimum separation criteria are satisfied both prior to and after the merge point under nominal conditions. This corresponds then to a nominal situation which requires no air traffic controller intervention. When the Relative Ground Speed errors are added, it is possible to determine if the resulting input variable estimate leads to an incorrect aircraft ordering or merging decision.

A convenient way to obtain two jet trajectories is to take the nominal jet trajectory of the previous two figures and assume that all the jet aircraft for that runway approach nominally flies the same distance histories. The only difference is in the time histories. Hence, it is convenient to use the same jet time history and obtain the second jet time history by shifting the nominal jet time history using a time bias. This approach yields the time histories for the two jets as shown in Figure 3-5.

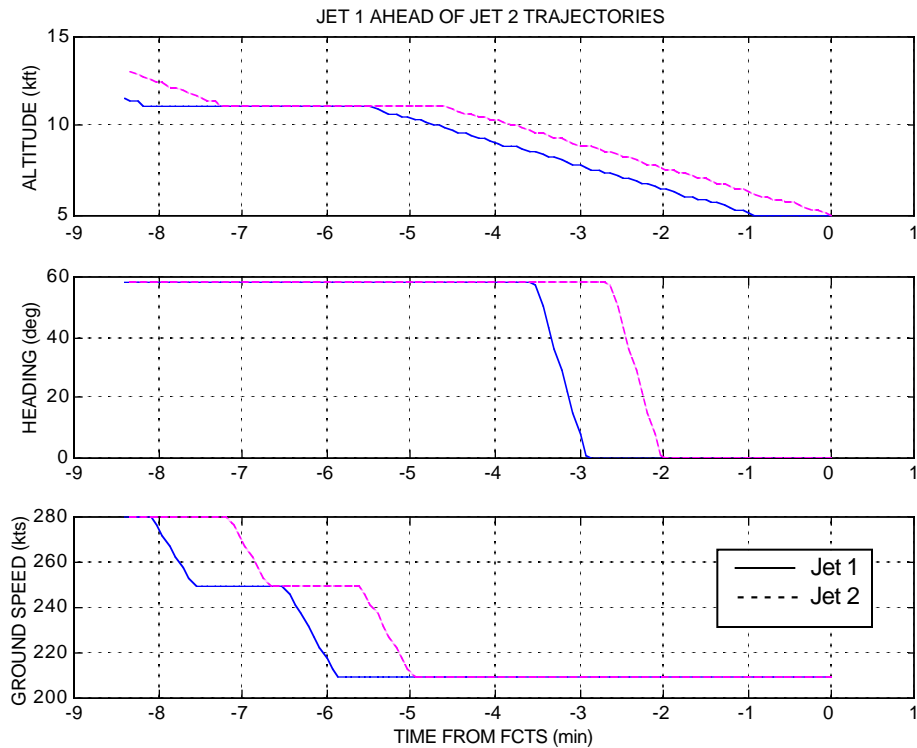


Figure 3-5. FAST TS Jet 1 Ahead of Jet 2 Time Histories

In this figure, the first Jet reaches the Downwind flight path segment approximately 6 minutes prior to t_{FCTS} while the second aircraft reaches it 5 minutes prior to t_{FCTS} . This time separation of one minute is the minimum separation for two jets. As noted earlier, t_{FCTS} is determined when the turboprop aircraft reaches the Downwind segment.

In addition, the nominal jet and turboprop trajectories are time shifted (biased) to produce a minimum separation merging scenario. This scenario corresponds to the Turboprop ahead of the Jet at FCTS, as illustrated in Figure 3-6. Under this scenario, the required minimum separation distance has been used. Hence, the Turboprop is ahead of the Jet by 3 nm. at FCTS.

3.3 Input Variable Errors

In this section, the error equations will be derived for those FAST SL Membership Function input variables which are sensitive to ground speed errors. These errors are also illustrated using the scenarios described in the last section.

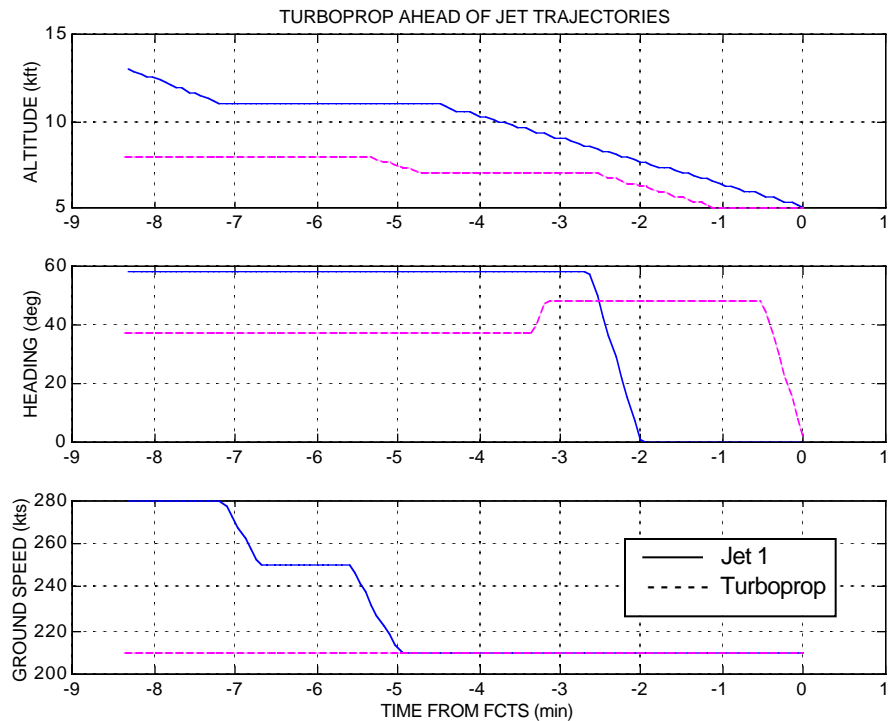


Figure 3-6. FAST TS Turboprop Ahead of Jet Time Histories

3.3.1 Aircraft Radar Tracking Errors

Before examining the input variable error statistics which affect the FAST SL performance, it is worthwhile to examine the fundamental aircraft radar tracking errors. In particular, since the focus of this study is on the effect of ground speed errors, consider how significant are the other tracking errors? These latter errors include the aircraft heading error and the ground path errors.

In Figures 3-7 and 3-8 are presented the ground speed error statistics respectively for the jet and the turboprop aircraft. Clearly any ground speed errors might lead to an unforeseen overtake situation which might lead to a possible collision. In the context of the Relative Ground Speed Propositions used by the Ordering and Merging Procedure of a GENERAL-Type Spatial Constraint, any Relative Ground Speed magnitude greater than 20 kts. is significant. Based on these two criteria and the ground speed error statistics of these two figures, the ground speed errors have to be considered significant, particularly during speed reduction maneuvers and heading changes.

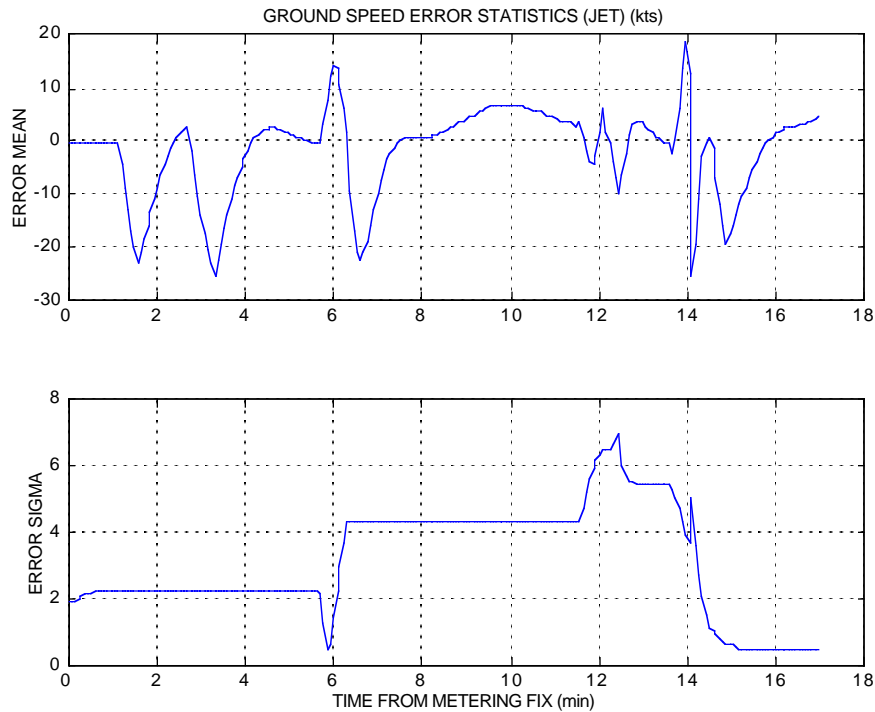


Figure 3-7. Jet Ground Speed Tracking Errors

In Figures 3-9 and 3-10 are presented the heading error statistics respectively for the jet and the turboprop aircraft. The heading mean errors are seen to vary between ± 25 deg. for these two cases. There are no criteria, however, for determining whether these heading errors are significant. This arises from the fact that none of the Propositions use relative heading directly or indirectly as an input. Hence, even though the heading errors are large, they can be ignored in an error analysis of the FAST SL input variables.

In Figure 3-11 and 3-12 are presented the ground path error statistics respectively for the jet and the turboprop aircraft. These errors appear to be bounded loosely between ± 0.1 nm. Since the minimum required separation distance for two aircraft is equal to 3 nm or greater, the ground path error statistics can be ignored.

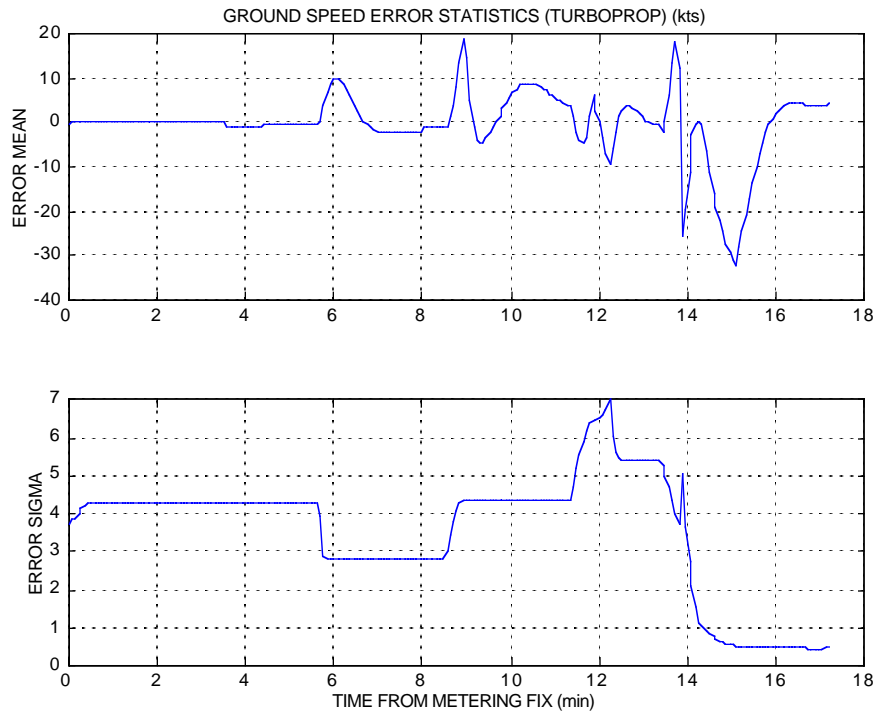


Figure 3-8. Turboprop Ground Speed Tracking Errors

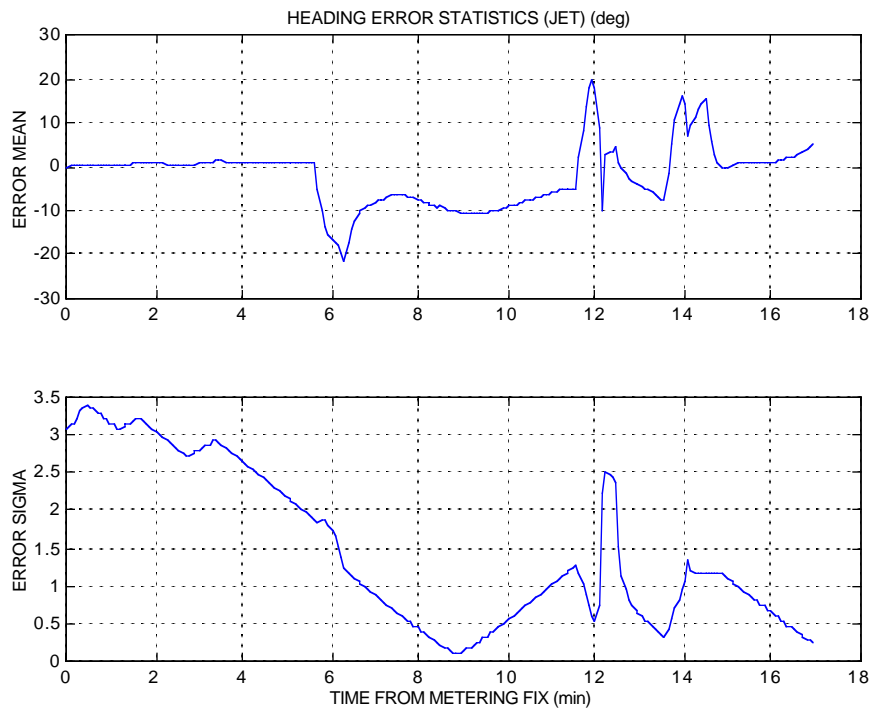


Figure 3-9. Jet Heading Tracking Errors

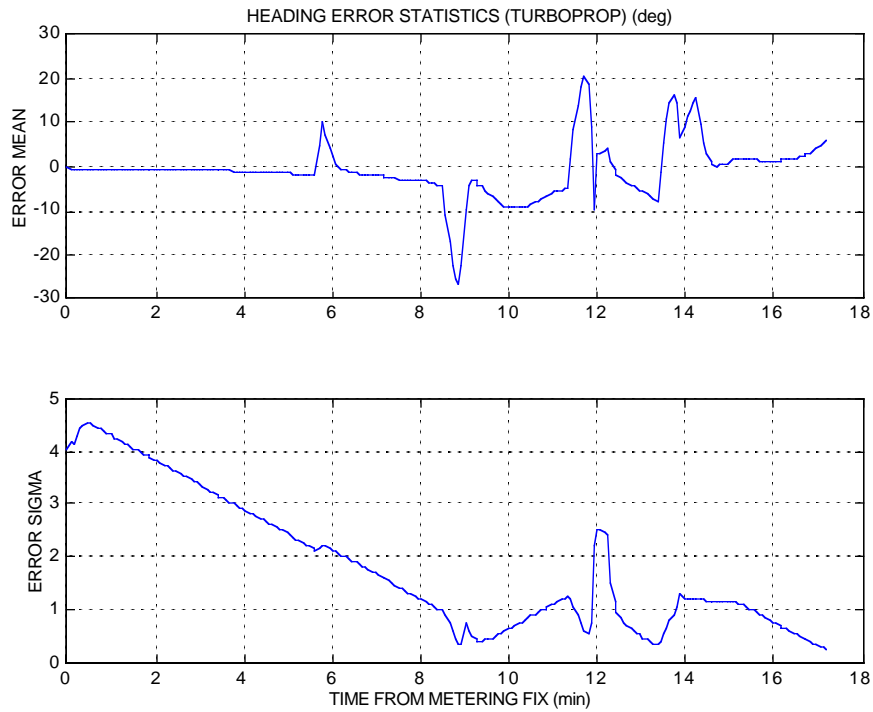


Figure 3-10. Turboprop Heading Tracking Errors

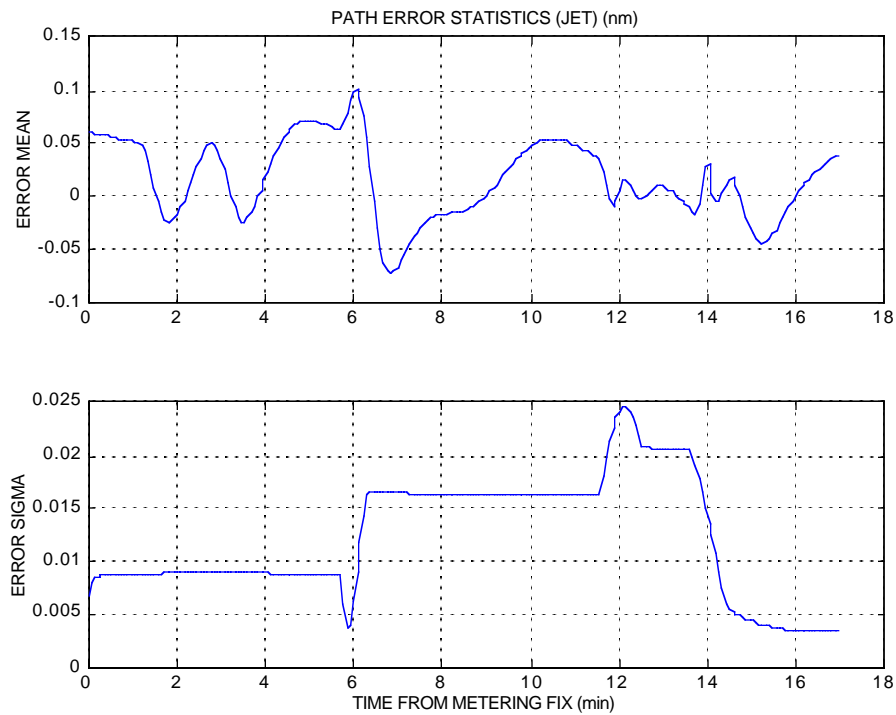


Figure 3-11. Jet Ground Path Tracking Errors

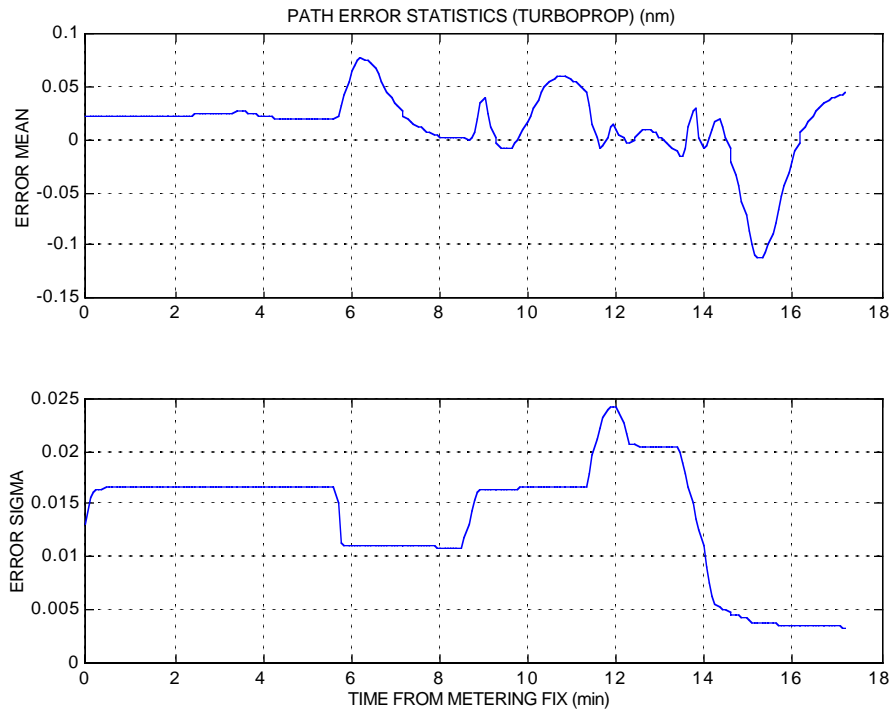


Figure 3-12. Turboprop Ground Path Tracking Errors

3.4.2 Relative Ground Speed

For aircraft A ahead of aircraft B, the Relative Ground Speed error is given by:

$$\delta V_{G,AB}(t_k) \equiv (\delta V_{G,A}(t_k) - \delta V_{G,B}(t_k))$$

where, V_G = current ground speed of aircraft

The Relative Ground Speed error at the current time is best illustrated for the two jet scenario which is shown in Figure 3-13. The Relative Ground Speed error at FCTS is best illustrated for the case which involves the jet and the turboprop aircraft illustrated in Figure 3-14. For the latter figure it was assumed that the current (prior to FCTS) Relative Ground Speed error will be the same error at FCTS. In each of these two figure is shown the nominal history on top. This is followed by the estimate history subplot with the ± 2 sigma bounds. Finally, the last two subplots show the mean and the standard deviation histories.

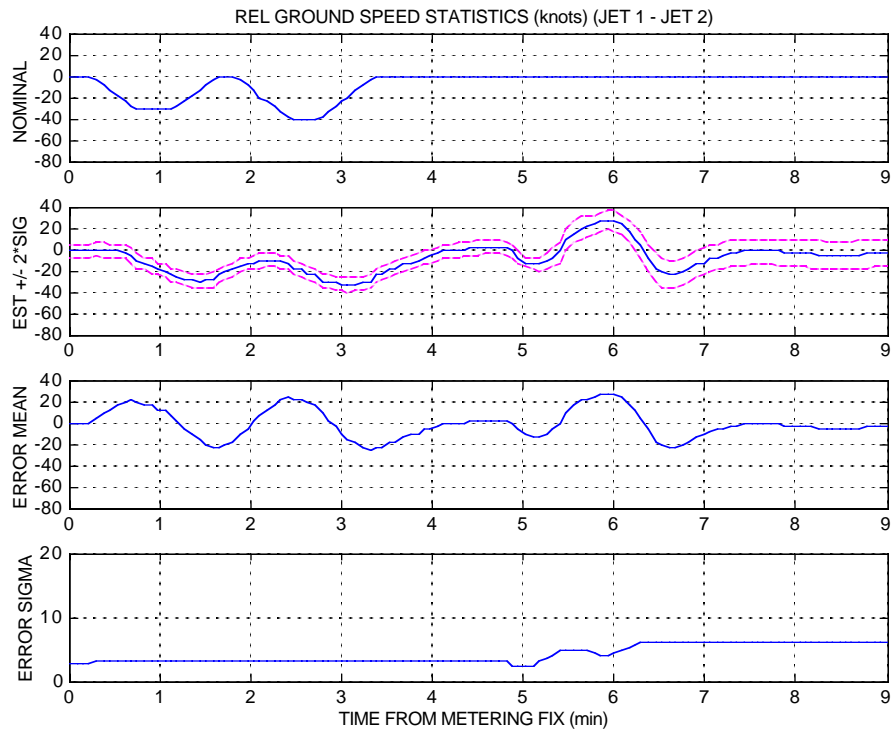


Figure 3-13. Relative Ground Speed Statistics (Jet 1 ahead of Jet 2)

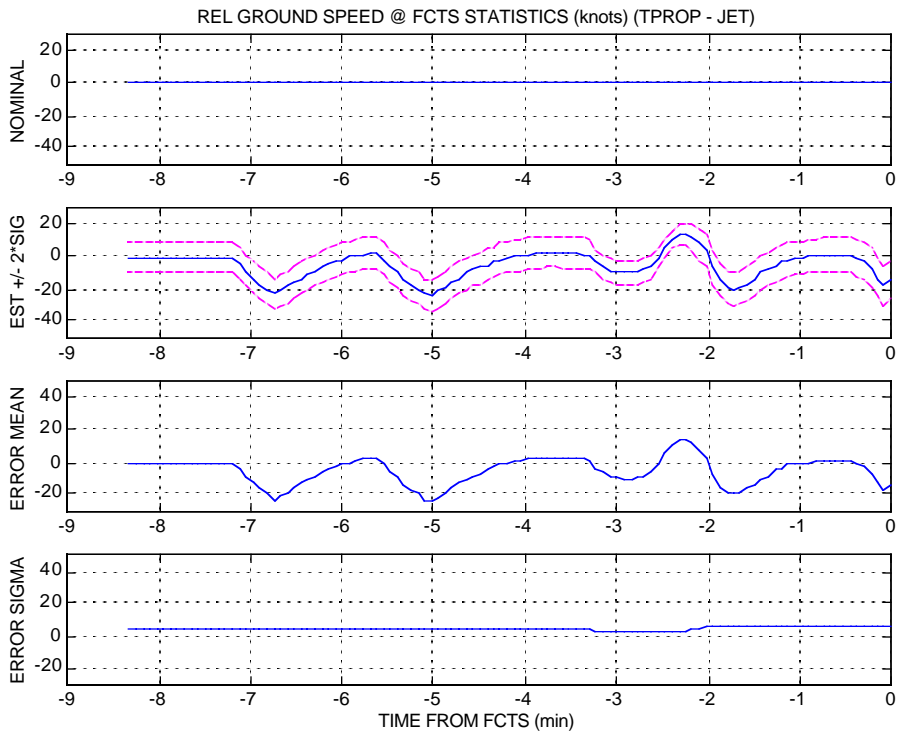


Figure 3-14. Relative Ground Speed at FCTS Statistics (Turboprop ahead of Jet)

The assumption that the Relative Ground Speed error at FCTS will be the same as the current Relative Ground Speed, needs to be examined more closely. At certain flight path distances from the runway threshold, the aircraft must adhere to three different speed limits. The first speed limit of a 250 knot ground speed occurs as soon as the initial descent is completed. The second speed limit of 210 knots occurs approximately mid-way along the Downwind path segment. The third speed limit is determined by aircraft type and occurs prior to reaching the outer marker. In general, if an aircraft has a ground speed less than the current speed limit, it is allowed to proceed with that speed to the next speed limit.

FAST TS considers the nominal speed reductions at the nominal locations in computing the predicted trajectory for an aircraft. It also allows the aircraft to proceed with its current estimate of the speed until the next speed reduction. Hence, normally the current ground speed estimation error will only propagate to the next speed limit.

Also, if the actual ground speed equals the current speed limit, the positive estimation errors is clipped if the air traffic controller tells the aircraft to slow down to remain within the stated speed limit because it is large enough. The negative estimation errors, will give the appearance that the aircraft is within the speed limit and hence are left unchanged until the next speed limit.

There are other sources of ground speed deviations from the nominal. These sources of ground speed deviation include the flight technical errors, and unpredicted winds. Hence, a conservative approach is to ignore the moderating effects of the ground speed limits, discussed above, on the ground speed estimation errors introduced through radar tracking.

3.4.3 Normalized Separation Distance at FCTS

The estimation error for the NSD_{FCTS} was derived in (Mueller, 1998). It was shown that Relative Ground Speed errors corrupt the estimates of the time of FCTS and thereby produce an error in the NSD_{FCTS} . The estimation error equation is given as:

$$\delta NSD_{FCTS}(t_k) = \begin{cases} \frac{(\delta d_{B,FCTS}(t_k) - \delta d_{A,FCTS}(t_k))}{\Delta d_{AB}}, & \text{if, } d_{B,FCTS}(t_k) > d_{A,FCTS}(t_k) \\ \frac{(\delta d_{B,FCTS}(t_k) - \delta d_{A,FCTS}(t_k))}{\Delta d_{BA}}, & \text{if, } d_{B,FCTS}(t_k) \leq d_{A,FCTS}(t_k) \end{cases}$$

or,

$$\delta NSD_{FCTS}(t_k) \equiv \left\{ \begin{array}{l} \left[\left((t_{FCTS,AB} - t_k) + I_{k,FCTS,A} (V_{G,A}(t_{FCTS}) - V_{G,B}(t_{FCTS})) \right) \delta V_{G,A}(t_k) - \right. \\ \left. \left[(t_{FCTS,AB} - t_k) + I_{k,FCTS,B} (V_{G,A}(t_{FCTS}) - V_{G,B}(t_{FCTS})) \right] \delta V_{G,B}(t_k) \right] / \Delta d_{AB}, \\ \text{if, A ahead of B at FCTS} \\ \\ \left[\left((t_{FCTS,AB} - t_k) + I_{k,FCTS,A} (V_{G,A}(t_{FCTS}) - V_{G,B}(t_{FCTS})) \right) \delta V_{G,A}(t_k) - \right. \\ \left. \left[(t_{FCTS,AB} - t_k) + I_{k,FCTS,B} (V_{G,A}(t_{FCTS}) - V_{G,B}(t_{FCTS})) \right] \delta V_{G,B}(t_k) \right] / \Delta d_{BA}, \\ \text{if, B ahead of A at FCTS} \end{array} \right.$$

where, $t_{FCTS,AB}$ = First Common Time Step (FCTS) for aircraft A and B

$$I_{FCTS,A,k} \equiv - \left[\int_{t_k}^{t_{FCTS}} \frac{dt}{V_{G,A}(t)} \right]$$

$$I_{FCTS,B,k} \equiv - \left[\int_{t_k}^{t_{FCTS}} \frac{dt}{V_{G,B}(t)} \right]$$

The minimum required separation distance, Δd_{nm} , between two aircraft A and B in the TRACON is presented in Table 2-5. This table shows that this separation distance depends on the size and sequence of the two aircraft which are in-track to each other.

The NSD_{FCTS} statistics histories for the Turboprop ahead of the Jet are presented in Figure 3-15. The nominal history is shown in the top subplot while the second subplot shows the estimate history and its ± 2 sigma bound. Finally, the last two subplots present the mean and standard deviation histories. The reason for the diminishing standard deviation error as time gets closer to the FCTS arises from the fact that the FCTS prediction is made over a shorter time interval.

3.4.4 Relative ETA Magnitude:

Since the relative ETA magnitude is a non-linear function, the corresponding error has to be obtained as follows:

Since, $\tau_{AB}(t_k) \equiv |TA_{B,Early}(t_k) - TA_{A,Early}(t_k)| = \sqrt{(TA_{B,Early}(t_k) - TA_{A,Early}(t_k))^2}$
then,

$$\delta \tau_{AB}(t_k) = \left| (TA_{B,Early}(t_k) + \delta TA_{B,Early}(t_k)) - (TA_{A,Early}(t_k) + \delta TA_{A,Early}(t_k)) \right| - |TA_{B,Early}(t_k) - TA_{A,Early}(t_k)|$$

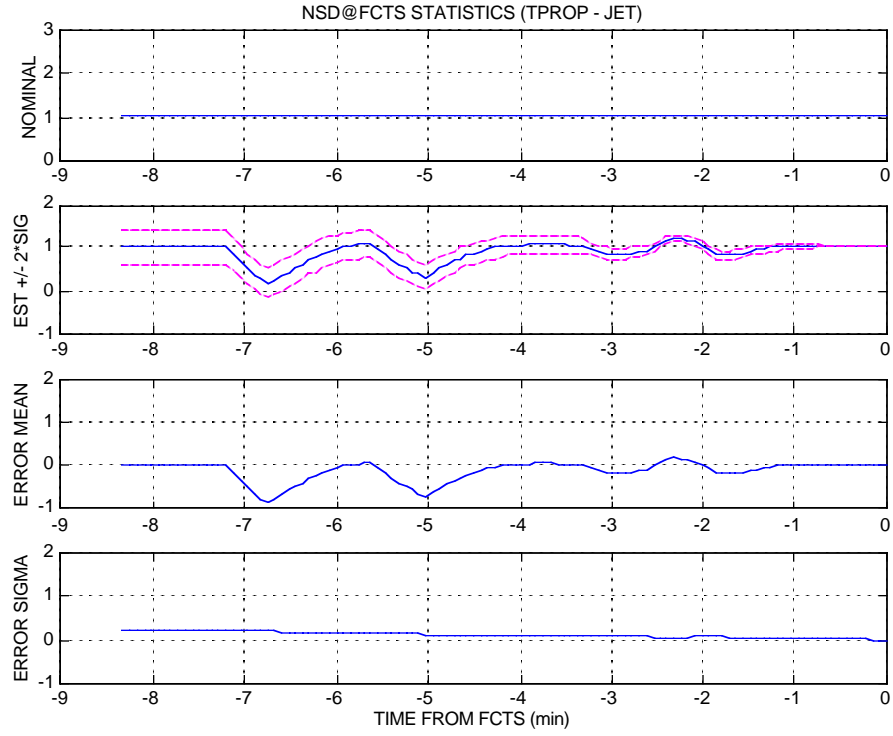


Figure 3-15. NSD at FCTS Statistics (Turboprop ahead of Jet)

or since,

$$\delta TA_{\text{Early}}(t_k) = - \left[I(t_k)_{TA, \text{Early}} \delta V_G(t_k) + \left(\frac{1}{V_G(t_k)} \right) \delta d(t_k) \right] \cong -I(t_k)_{TA, \text{Early}} \delta V_G(t_k)$$

then,

$$\delta \tau_{AB}(t_k) \cong \left| \left(TA_{B, \text{Early}}(t_k) - I(t_k)_{TA, \text{Early}, B} \delta V_{G, B}(t_k) \right) - \left(TA_{A, \text{Early}}(t_k) - I(t_k)_{TA, \text{Early}, A} \delta V_{G, A}(t_k) \right) \right| - \left| TA_{B, \text{Early}}(t_k) - TA_{A, \text{Early}}(t_k) \right|$$

Now if: $|\delta TA_{A, \text{Early}}(t_k)| \ll |TA_{A, \text{Early}}(t_k)|$, and, $|\delta TA_{B, \text{Early}}(t_k)| \ll |TA_{B, \text{Early}}(t_k)|$
then,

$$\delta \tau_{AB}(t_k) \cong \left(\delta TA_{B, \text{Early}}(t_k) - \delta TA_{A, \text{Early}}(t_k) \right) \cdot \text{Sign} \left\{ TA_{B, \text{Early}}(t_k) - TA_{A, \text{Early}}(t_k) \right\}$$

or,

$$\delta \tau_{AB}(t_k) \cong \left(I(t_k)_{TA, \text{Early}, A} \delta V_{G, A}(t_k) - I(t_k)_{TA, \text{Early}, B} \delta V_{G, B}(t_k) \right) \cdot \text{Sign} \left\{ TA_{B, \text{Early}} - TA_{A, \text{Early}} \right\}$$

where,

$$I(t_k)_{TA,Early,A} \equiv - \left[\int_{t_k}^{TA_{A,Early}} \frac{dt}{V_{G,A}(t)} \right], \text{ and } I(t_k)_{TA,Early,B} \equiv - \left[\int_{t_k}^{TA_{B,Early}} \frac{dt}{V_{G,B}(t)} \right]$$

The Relative ETA Magnitude statistics are illustrated in Figure 3-16 for the two jets. The second subplot also shows the 2 sigma bounds around the estimate.

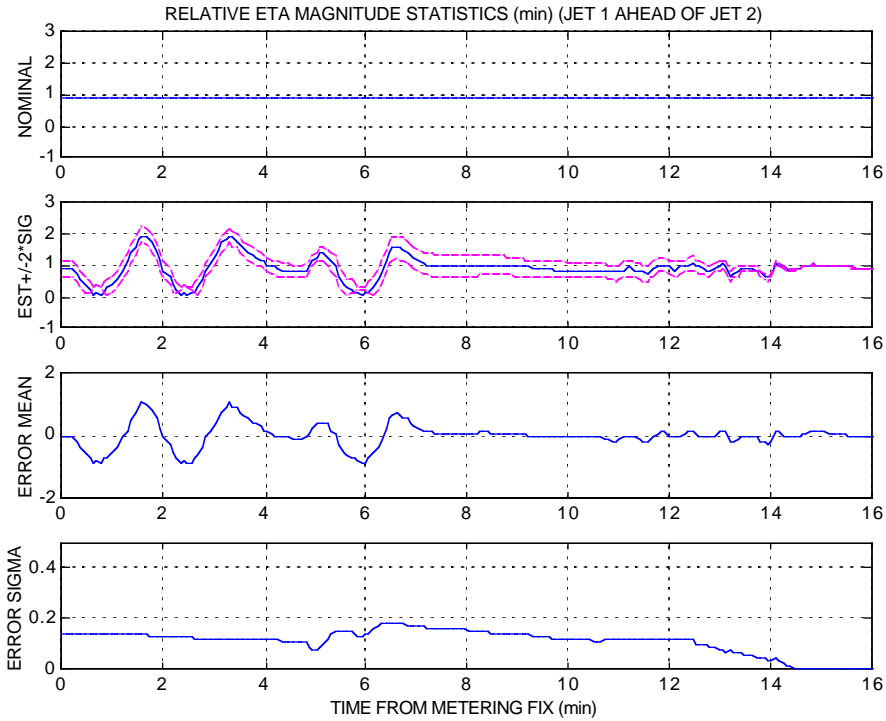


Figure 3-16. Relative ETA Magnitude Statistics (Jet 1 ahead of Jet 2)

3.4.5 Controllability

$$\delta\kappa \equiv \begin{cases} \frac{(\delta TA_{A,Late} - \delta STA_{A,BA})}{\Delta t_{BA}}, & \text{if A is sequenced behind B} \\ -\frac{(\delta TA_{B,Late} - \delta STA_{B,AB})}{\Delta t_{AB}}, & \text{if B is sequenced behind A} \end{cases}$$

or since,

$$\delta TA_{Late}(t_k) \equiv -I(t_k)_{TA,Late} \delta V_{G,Late}(t_k)$$

and,

$$\delta STA(t_k) \equiv -I(t_k)_{STA} \delta V_G(t_k)$$

then,

$$\delta\kappa(t_k) \equiv \begin{cases} \frac{I(t_k)_{STA,A,BA} \delta V_{G,A}(t_k) - I(t_k)_{TA,Late,A} \delta V_{G,A,Late}(t_k)}{\Delta t_{BA}}, & \text{if A is sequenced behind B} \\ -\frac{I(t_k)_{STA,B,AB} \delta V_{G,B}(t_k) - I(t_k)_{TA,Late,B} \delta V_{G,B,Late}(t_k)}{\Delta t_{AB}}, & \text{if B is sequenced behind A} \end{cases}$$

where, Δt_{nm} = minimum required separation time with aircraft n ahead of m

$$I(t_k)_{STA} \equiv -\left[\int_{t_k}^{STA} \frac{dt}{V_G(t)} \right], \text{ and } I(t_k)_{TA,Late} \equiv -\left[\int_{t_k}^{TA_{Late}} \frac{dt}{V_G(t)} \right]$$

The minimum required separation time between two aircraft was previously shown in Table 2-4. The Controllability is illustrated for the two jet case in Figure 3-17.

3.4.6 Excess Delay

Like the ETA magnitude, the excess delay is another non-linear function for which the error has to be obtained as follows:

Since,

$$\varsigma \equiv 100 \cdot \begin{cases} \left(\frac{TA_{A,Late} - STA_{A,BA}}{TA_{A,Late} - TA_{A,Early}} \right), & \text{if A is sequenced behind B} \\ -\left(\frac{TA_{B,Late} - STA_{B,AB}}{TA_{B,Late} - TA_{B,Early}} \right), & \text{if B is sequenced behind A} \end{cases}$$

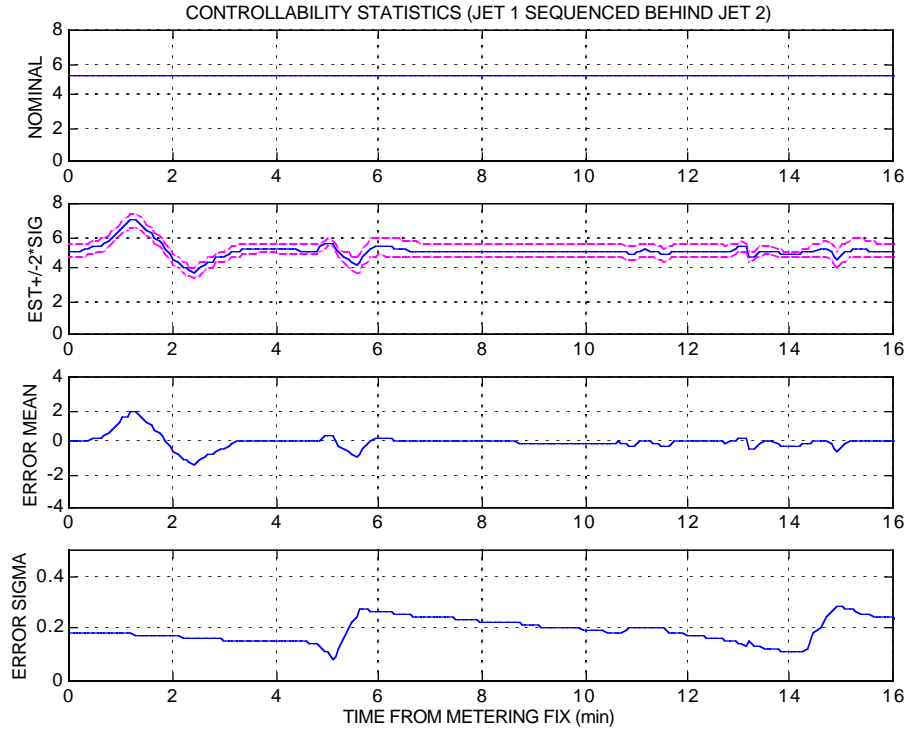


Figure 3-17. Controllability Statistics (Jet 1 ahead of Jet 2)

then,

$$\delta\zeta \equiv 100 \cdot \begin{cases} \left(\frac{(TA_{A,Late} + \delta TA_{A,Late}) - (STA_{A,BA} + \delta STA_{A,BA})}{(TA_{A,Late} + \delta TA_{A,Late}) - (TA_{A,Early} + \delta TA_{A,Early})} \right) - \left(\frac{TA_{A,Late} - STA_{A,BA}}{TA_{A,Late} - TA_{A,Early}} \right), & \text{if A is sequenced behind B} \\ - \left(\frac{(TA_{B,Late} + \delta TA_{B,Late}) - (STA_{B,AB} + \delta STA_{B,AB})}{(TA_{B,Late} + \delta TA_{B,Late}) - (TA_{B,Early} + \delta TA_{B,Early})} \right) + \left(\frac{TA_{B,Late} - STA_{B,AB}}{TA_{B,Late} - TA_{B,Early}} \right), & \text{if B is sequenced behind A} \end{cases}$$

or since,

$$\delta TA_{Early}(t_k) \cong -I(t_k)_{TA,Early} \delta V_{G,Early}(t_k)$$

$$\delta TA_{Late}(t_k) \cong -I(t_k)_{TA,Late} \delta V_{G,Late}(t_k)$$

and,

$$\delta STA(t_k) \cong -I(t_k)_{STA} \delta V_G(t_k)$$

with,

$$I(t_k)_{STA} \equiv - \left[\int_{t_k}^{STA} \frac{dt}{V_G(t)} \right]$$

$$I(t_k)_{TA,Early} \equiv - \left[\int_{t_k}^{TA_{Early}} \frac{dt}{V_{G,Early}(t)} \right]$$

$$I(t_k)_{TA,Late} \equiv - \left[\int_{t_k}^{TA_{Late}} \frac{dt}{V_{G,Late}(t)} \right]$$

$$\delta\zeta \equiv 100 \cdot \left\{ \begin{array}{l} \left(\frac{\left(TA_{A,Late} - I(t_k)_{TA,Late,A} \delta V_{G,A,Late}(t_k) \right) - \left(STA_{A,BA} - I(t_k)_{STA,A} \delta V_{G,A}(t_k) \right)}{\left(TA_{A,Late} - I(t_k)_{TA,Late,A} \delta V_{G,A,Late}(t_k) \right) - \left(TA_{A,Early} - I(t_k)_{TA,Early,A} \delta V_{G,A,Early}(t_k) \right)} \right. \\ \left. - \left(\frac{TA_{A,Late} - STA_{A,BA}}{TA_{A,Late} - TA_{A,Early}} \right), \quad \text{if A is sequenced behind B} \right. \\ \left. - \left(\frac{\left(TA_{B,Late} - I(t_k)_{TA,Late,B} \delta V_{G,B,Late}(t_k) \right) - \left(STA_{B,AB} - I(t_k)_{STA,B} \delta V_{G,B}(t_k) \right)}{\left(TA_{B,Late} - I(t_k)_{TA,Late,B} \delta V_{G,B,Late}(t_k) \right) - \left(TA_{B,Early} - I(t_k)_{TA,Early,B} \delta V_{G,B,Early}(t_k) \right)} \right) \right. \\ \left. + \left(\frac{TA_{B,Late} - STA_{B,AB}}{TA_{B,Late} - TA_{B,Early}} \right), \quad \text{if B is sequenced behind A} \right\}$$

Now, if : $|\delta TA_{Early}| \ll |TA_{Early}|$, $|\delta TA_{Late}| \ll |TA_{Late}|$, and, $|\delta STA| \ll |STA|$,
then,

$$\delta\zeta \equiv \left\{ \begin{array}{l} \frac{100}{\left(TA_{A,Late} - TA_{A,Early} \right)^2} \left[\left(TA_{A,Late} - STA_{A,BA} \right) \delta TA_{A,Early} \right. \\ \left. - \left(TA_{A,Early} - STA_{A,BA} \right) \delta TA_{A,Late} - \left(TA_{A,Late} - TA_{A,Early} \right) \delta STA_{A,BA} \right], \\ \quad \text{for A sequenced behind B} \\ \frac{-100}{\left(TA_{B,Late} - TA_{B,Early} \right)^2} \left[\left(TA_{B,Late} - STA_{B,AB} \right) \delta TA_{B,Early} \right. \\ \left. - \left(TA_{B,Early} - STA_{B,AB} \right) \delta TA_{B,Late} - \left(TA_{B,Late} - TA_{B,Early} \right) \delta STA_{B,AB} \right], \\ \quad \text{for B sequenced behind A} \end{array} \right.$$

or,

$$\delta\zeta(t_k) \equiv \begin{cases} \frac{-100}{\left(TA_{A,Late}(t_k) - TA_{A,Early}(t_k)\right)^2} \left[\begin{aligned} &\left(TA_{A,Late}(t_k) - STA_{A,BA}(t_k)\right)I(t_k)_{TA,Early,A} \delta V_{G,A,Early}(t_k) \\ &- \left(TA_{A,Early}(t_k) - STA_{A,BA}(t_k)\right)I(t_k)_{TA,Late,A} \delta V_{G,A,Late}(t_k) \\ &- \left(TA_{A,Late}(t_k) - TA_{A,Early}(t_k)\right)I(t_k)_{STA,A} \delta V_{G,A}(t_k) \end{aligned} \right], \\ \text{for A sequenced behind B} \\ \\ \frac{100}{\left(TA_{B,Late}(t_k) - TA_{B,Early}(t_k)\right)^2} \left[\begin{aligned} &\left(TA_{B,Late}(t_k) - STA_{B,AB}(t_k)\right)I(t_k)_{TA,Early,B} \delta V_{G,B,Early}(t_k) \\ &- \left(TA_{B,Early}(t_k) - STA_{B,AB}(t_k)\right)I(t_k)_{TA,Late,B} \delta V_{G,B,Late}(t_k) \\ &- \left(TA_{B,Late}(t_k) - TA_{B,Early}(t_k)\right)I(t_k)_{STA,B} \delta V_{G,B}(t_k) \end{aligned} \right], \\ \text{for B sequenced behind A} \end{cases}$$

The Excess Delay statistics are illustrated for the two jet case in Figure 3-18.

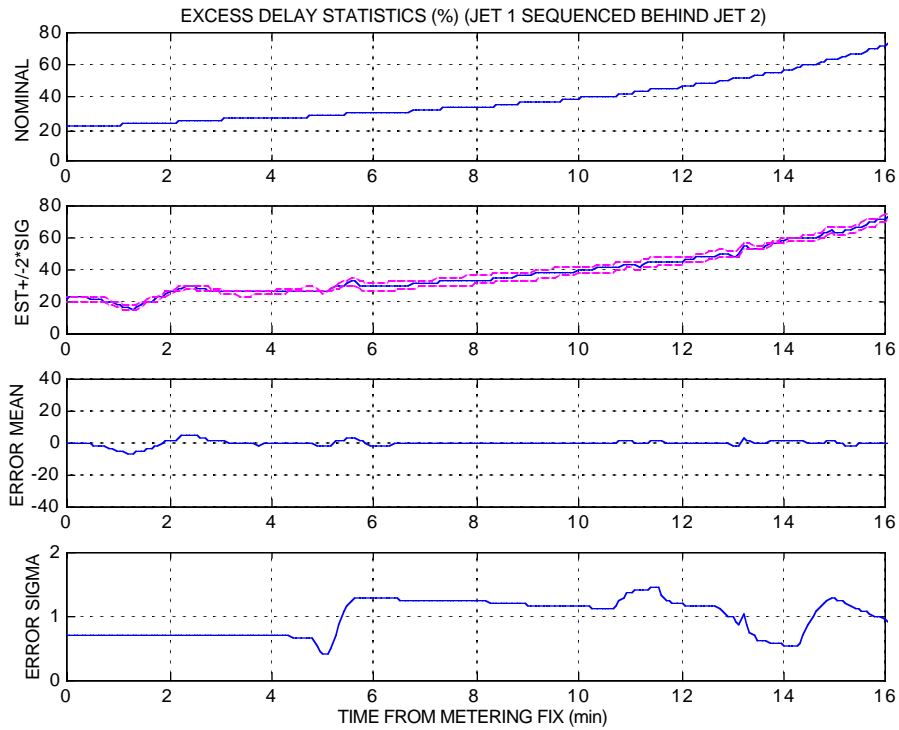


Figure 3-18. Excess Delay Statistics (Jet 1 ahead of Jet 2)

3.4.7 Normalized Delay Savings

$$\delta NDS = \frac{(\delta STA_{B,BA} - \delta TA_{B,Early}) + (\delta STA_{A,BA} - \delta TA_{A,Early})}{\Delta t_{BA}} - \frac{(\delta STA_{A,AB} - \delta TA_{A,Early}) + (\delta STA_{B,AB} - \delta TA_{B,Early})}{\Delta t_{AB}}$$

$$\text{or, } \delta NDS = \left(\frac{1}{\Delta t_{AB}} - \frac{1}{\Delta t_{BA}} \right) (\delta TA_{A,Early} + \delta TA_{B,Early}) - \left(\frac{1}{\Delta t_{AB}} \right) \delta STA_{A,AB} + \left(\frac{1}{\Delta t_{BA}} \right) \delta STA_{A,BA} - \left(\frac{1}{\Delta t_{AB}} \right) \delta STA_{B,AB} + \left(\frac{1}{\Delta t_{BA}} \right) \delta STA_{B,BA}$$

then,

$$\begin{aligned} \delta NDS(t_k) = & - \left(\frac{1}{\Delta t_{AB}} - \frac{1}{\Delta t_{BA}} \right) \left(I(t_k)_{TA,Early,A} \delta V_{G,A,Early}(t_k) \right. \\ & + I(t_k)_{TA,Early,B} \delta V_{G,B,Early}(t_k) \Big) \\ & + \left(\frac{I(t_k)_{STA,A,AB}}{\Delta t_{AB}} \right) \delta V_{G,A,AB}(t_k) - \left(\frac{I(t_k)_{STA,A,BA}}{\Delta t_{BA}} \right) \delta V_{G,A,BA}(t_k) \\ & + \left(\frac{I(t_k)_{STA,B,AB}}{\Delta t_{AB}} \right) \delta V_{G,B,AB}(t_k) - \left(\frac{I(t_k)_{STA,B,BA}}{\Delta t_{BA}} \right) \delta V_{G,B,BA}(t_k) \end{aligned}$$

$$\text{since, } \delta TA_{Early}(t_k) \cong -I(t_k)_{TA,Early} \delta V_{G,Early}(t_k)$$

$$\text{and, } \delta STA(t_k) \cong -I(t_k)_{STA} \delta V_G(t_k)$$

$$\text{with, } I(t_k)_{STA} \cong - \left[\int_{t_k}^{STA} \frac{dt}{V_G(t)} \right]$$

$$I(t_k)_{TA,EARLY} \cong - \left[\int_{t_k}^{TA_{EARLY}} \frac{dt}{V_{G,EARLY}(t)} \right]$$

where,

$STA_{n,nm}$ = scheduled time of arrival of aircraft n, given a sequence of nm

$STA_{m,nm} = \max\{ STA_{n,nm} + \Delta t_{nm}, TA_{m,Early} \}$, with $STA_{1stA/C} = TA_{1st,Early}$

$TA_{n,Early}, TA_{n,Late}$ = earliest and latest time of arrival for aircraft n

Δt_{nm} = required separation time between aircraft given a sequence of nm

The earliest time of arrival, $TA_{n,Early}$, is obtained by determining the time it takes an aircraft to reach the runway threshold when it takes the shortest flight path and the latest speed reductions. The latest time of arrival, $TA_{n,Late}$, is obtained by determining the time required for an aircraft to reach the runway threshold when it takes the longest flight path, using any flight path extensions, and the earliest speed reductions.

The Normalized Delay Savings statistics are illustrated in Figure 3-19 for the two jet case.

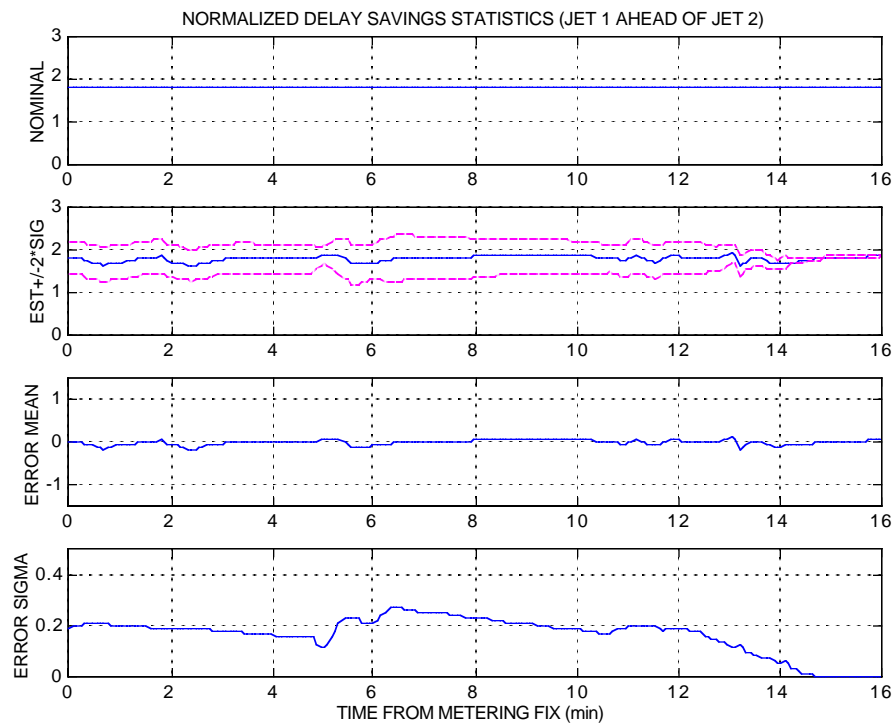


Figure 3-19. Normalized Delay Savings Statistics (Jet 1 ahead of Jet 2)

4.0 FIGURES OF MERIT

In this chapter figures of merit are defined for those FAST SL input variables which are sensitive to ground speed errors. These figures of merit are then be used to establish the performance of the Propositions which are affected by these input variable errors. The intent is to examine how uncertainties in the Proposition input variable might lead to incorrect decisions reached by that Proposition and the associated Procedure of which it is a part.

The approach which is used consists of a general review of several figures of merit and a comparison of their individual merits. This also involves the derivation of the equations for four new figures of merit. In the process, the probability density function for each ground speed dependent input variable is introduced. To evaluate the preferred figures of merit, parameters is provided for these figures of merit.

4.1 Relative Ground Speed Figure of Merit

The Relative Ground Speed Membership and Consequent Function pair is illustrated in Figure 4-1.

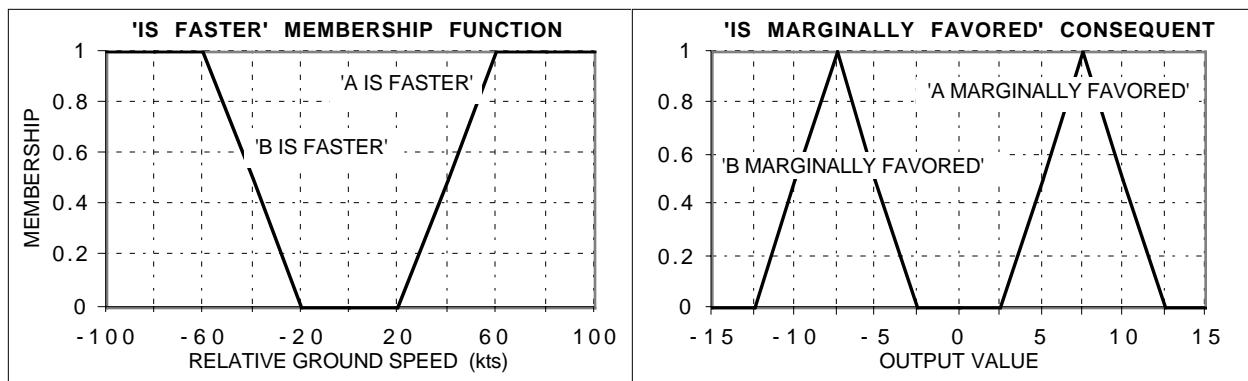


Figure 4-1. Relative Ground Speed Membership and Consequent Functions

It is worthwhile to examine this figure in greater detail in order to understand how it is used. If two aircraft, A and B, are positioned such that A is ahead of B (AB), then if A is faster than B, the Relative Ground Speed is positive. If A is slower than B than the Relative Ground Speed is negative. For these two aircraft at a particular time, only one or the other outcome is possible, given their current relative position.

Based on this figure and the uncertainties in the Relative Ground Speed estimate, a number of error cases might occur. These are summarized in Table 4-1 and are illustrated in Figures 4-2 through 4-4.

Table 4-1 Relative Ground Speed Decision Error Cases

| Case | Actual Relative Ground Speed, $\Delta V_{G,A}$ | Estimated Relative Ground Speed, ΔV_G | Selected Output | Required Output |
|------|---|---|-----------------|-----------------|
| 1a | $-20 \text{ kts} < \Delta V_{G,A} < 20 \text{ kts}$ | $\Delta V_G < -20 \text{ kts}$ | -7.5 | 0 |
| 1b | | $\Delta V_G > 20 \text{ kts}$ | 7.5 | |
| 2a | $\Delta V_{G,A} < -20 \text{ kts}$ | $-20 \text{ kts} < \Delta V_G < 20 \text{ kts}$ | 0 | -7.5 |
| 2b | | $\Delta V_G > 20 \text{ kts}$ | 7.5 | |
| 3a | $\Delta V_{G,A} > 20 \text{ kts}$ | $-20 \text{ kts} < \Delta V_G < 20 \text{ kts}$ | 0 | 7.5 |
| 3b | | $\Delta V_G < -20 \text{ kts}$ | -7.5 | |

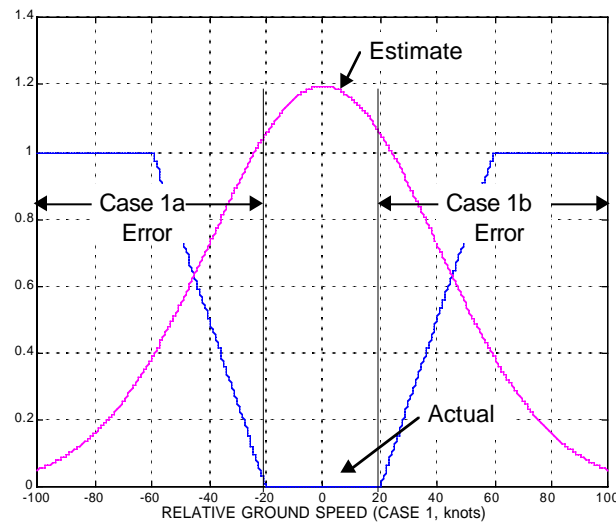


Figure 4-2. Figure of Merit for Relative Ground Speed (Case 1)

Note that there are three actions which can be taken, based on the Output of this Proposition: no action (Output = 0), keep the current order of the two aircraft (output is 7.5), or reverse the order of the two aircraft (Output = -7.5).

In these figures, it is assumed that the estimated Relative Ground Speed has a Gaussian probability density function (pdf). Furthermore, this pdf has a mean whose value is the actual Relative Ground Speed. Hence, for illustration purposes, the estimation error mean (difference between estimated and actual) is zero. Also, the magnitude of the pdf in these figures has been arbitrarily scaled up for clarity.

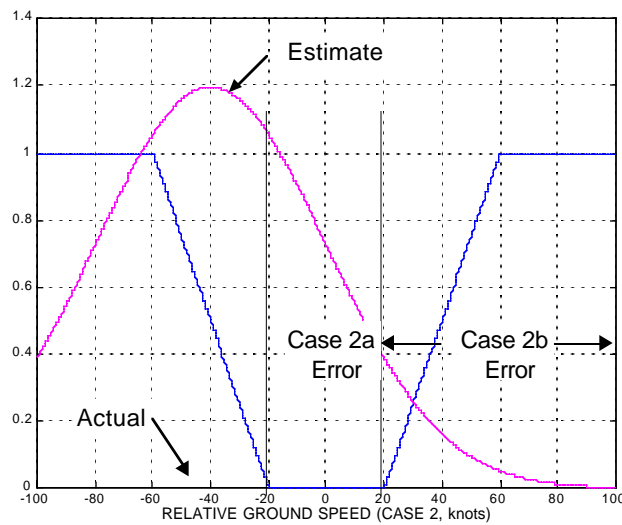


Figure 4-3. Figure of Merit for Relative Ground Speed (Case 2)

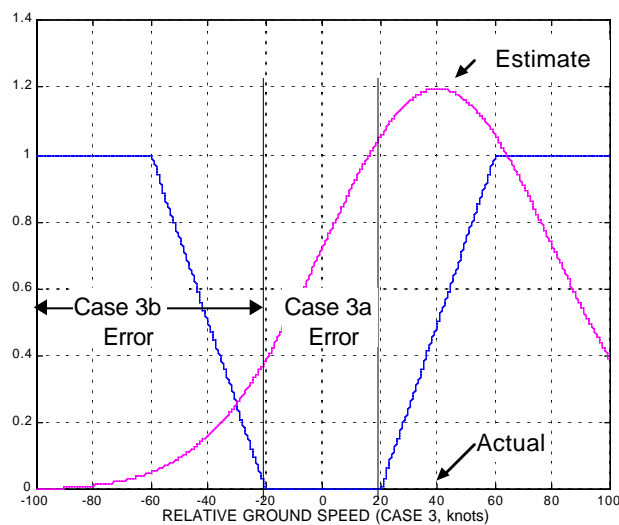


Figure 4-4. Figure of Merit for Relative Ground Speed (Case 3)

If the actual Relative Ground Speed is within the deadband of -20 knots to 20 knots, then Relative Ground Speed estimates less than -20 knots or greater than 20 knots, leads to an action when no action is required. A second case is where the actual Relative Ground Speed is less than -20 knots, but the estimated Relative Ground Speed lies within the deadband or is greater than 20 knots. Finally, a third case is where the actual Relative Ground Speed is greater than 20 knots, but the estimated Relative Ground Speed falls within the deadband or is less than -20 knots.

The decision error cases of Table 4-1 can be quantified using the Relative Ground Speed pdf.

$$f_{\Delta V_G}(\hat{\Delta V}_G) = \frac{1}{\sqrt{2\pi}\sigma_{\Delta V_G}} e^{-\left(\frac{(\hat{\Delta V}_G - \mu_{\Delta V_G})^2}{2\sigma_{\Delta V_G}^2}\right)}$$

$$\text{where, } \sigma_{\Delta V_G} \equiv \sqrt{\sigma_{V_{G,A}}^2 + \sigma_{V_{G,B}}^2}$$

$$\mu_{\Delta V_G} \equiv (\mu_{V_{G,A}} - \mu_{V_{G,B}})$$

Now the general Relative Ground Speed, ΔV_G , decision error probability (figure of merit) calculation is:

$$P_{\text{Case}} = \frac{1}{\sqrt{2\pi}\sigma_{\Delta V_G}} \int_{\Delta \hat{V}_{G,\text{Low}}}^{\Delta \hat{V}_{G,\text{Up}}} e^{-\left(\frac{(\hat{\Delta V}_G - \mu_{\Delta V_G})^2}{2\sigma_{\Delta V_G}^2}\right)} d(\hat{\Delta V}_G),$$

when, $\Delta V_{G,\text{Low}} < \Delta V_G < \Delta V_{G,\text{Up}}$

where, ΔV_G = actual Relative Ground Speed
 $\hat{\Delta V}_G$ = estimate of Relative Ground Speed

The parameters required to perform this integration for each case is presented in Table 4-2.

Table 4-2 ΔV_G Figure of Merit Parameters

| Case | $\Delta V_{G,\text{Low}}$ | $\Delta V_{G,\text{Up}}$ | $\Delta \hat{V}_{G,\text{Low}}$ | $\Delta \hat{V}_{G,\text{Up}}$ |
|----------|---------------------------|--------------------------|---------------------------------|--------------------------------|
| 1a 1b | -20 | 20 | $-\infty$ 20 | -20 ∞ |
| 2a 2b | $-\infty$ | -20 | -20 20 | 20 ∞ |
| 3a 3b | 20 | ∞ | -20 $-\infty$ | 20 -20 |

The above figure of merit was previously developed in (Mueller, 1998). The difficulty with this figure of merit is that it does not directly relate to the decision reached by this Proposition. In addition, it can lead to extensive tables, such as Table 4-1, for the

remaining Propositions which are sensitive to Relative Ground Speed errors. In other words, just by cataloging the different decision error cases, nothing new is learned about their impact on the decision reached by that Proposition.

4.2 New Figures of Merit

More relevant and practical figures of merit include the expected (mean) Output and Firing Strength. The Firing Strength is defined as the weighted output -- the product of the Output and its associated Weight. The Procedure determines its decision for the relative order of two aircraft by combining the individual Proposition Firing Strengths in a weighted sense, based on the degree of membership. Hence the expected Proposition Membership and Weight are two additional figures of merit.

To quantify these new figures of merit, consider the input x_n to Proposition n . Let $M_n(x_n)$ be the Proposition Membership, $O_n(x_n)$ the Proposition Output, $W_n(x_n)$ the Proposition Weight, and $S_n(x_n)$ the Proposition Firing Strength. Then if $f_{x_n}(x_n)$ is the probability density function of x_n , the expected (mean) Membership, μ_{M_n} , the expected (mean) Output, μ_{O_n} , the expected (mean) Weight, μ_{W_n} , and the expected (mean) Firing Strength, μ_{S_n} , are obtained, respectively, as follows:

$$\begin{aligned}\mu_{M_n} &\equiv \int_{-\infty}^{\infty} M_n(x_n) f_{x_n}(x_n) dx_n, \\ \mu_{O_n} &\equiv \int_{-\infty}^{\infty} O_n(x_n) f_{x_n}(x_n) dx_n, \\ \mu_{W_n} &\equiv \int_{-\infty}^{\infty} W_n(x_n) f_{x_n}(x_n) dx_n, \quad \text{and,} \\ \mu_{S_n} &\equiv \int_{-\infty}^{\infty} S_n(x_n) f_{x_n}(x_n) dx_n\end{aligned}$$

$$\text{where, } S_n(x_n) \equiv W_n(x_n) \cdot O_n(x_n)$$

In addition, the Procedure combines the Firing Strengths from all N Propositions which are included in this Procedure to obtain a weighted Firing Strength. This Procedure weighted Firing Strength determines whether the current order or the reverse order of two aircraft should be used. This decision is determined by examining the polarity (positive or negative) of this Procedure Normalized Firing Strength. Hence, if the Procedure Normalized Firing Strength is less than -7.5, the order of the aircraft is reversed. The Procedure Normalized Firing Strength is computed as follows:

$$S_{WP} \equiv \frac{\left(\sum_{n=1}^N S_n(x_n) \right)}{\left(\sum_{n=1}^N W_n(x_n) \right)}$$

Using this Procedure Normalized Firing Strength and the probability density functions for all N Proposition input variables, the mean Procedure Normalized Firing Strength is obtained as follows (Papoulis, 1965):

$$\begin{aligned} \mu_{S_{WP}} &\equiv \int_{-\infty}^{\infty} S_{WP} f_{S_{WP}}(S_{WP}) dS_{WP} \\ \mu_{S_{WP}} &= \int_{-\infty}^{\infty} \dots \int_{-\infty}^{\infty} \frac{\left(\sum_{n=1}^N S_n(x_n) \right)}{\left(\sum_{n=1}^N W_n(x_n) \right)} f_{x_1}(x_1) dx_1 \dots f_{x_N}(x_N) dx_N \end{aligned}$$

In the last expression, it is assumed that the x_n are independent. This assumption is only partially true since each x_n depends on some scaled version of the Relative Ground Speed error for the STA trajectories, as well as possibly for the Early or Late trajectories, of aircraft A or B.

On closer inspection, the last expression is fairly complicated. While the numerator can be expanded into the sum of N integrals, one for each Proposition, they all have the same common denominator, the sum of the N individual Weights. If the Weights were not dependent on x_n and were constant, then it can be shown that the above expression leads to the scaled sum of N Firing Strength means.

This last observation suggests the following approximation for the expected Procedure Normalized Firing Strength:

$$\mu_{S_{WP}} \cong \frac{\left(\sum_{n=1}^N \mu_{S_n} \right)}{\left(\sum_{n=1}^N \mu_{W_n} \right)}$$

Hence, this approximation uses the sum of the N mean Proposition Firing Strengths normalized by the sum of the N mean Proposition weights.

Additional analysis is required to determine how closely the statistical approximation in the last equation matches the precise definition for the Procedure Normalized Firing Strength. If this approximation is found not to be a good statistical approximation, it is still possible to use it as another figure of merit.

A heuristic argument to justify this approximation is as follows. In computing the n'th mean probability integral, the denominator is probably fairly constant. This is based on the fact that for the n'th integral, only the n'th Weight term is varying while the remaining (N-1) Weight terms are constant. Hence the Procedure Normalized Firing Strength is approximately the Procedure Firing Strength times a constant. This heuristic argument provides a reasonable justification for the use of the sum of the N mean Procedure Firing Strengths.

It is now possible to derive algorithms for the new figures of merit, defined above, for a generic Proposition as shown in Chapter 2. If x is a general input to a trapezoidal Membership Function, M(x):

$$M(x) = \begin{cases} 1, & \text{for, } x < b_L \\ \frac{(x - a_L)}{(b_L - a_L)}, & \text{for, } b_L \leq x < a_L \\ 0, & \text{for, } a_L \leq x < a_R \\ \frac{(x - a_R)}{(b_R - a_R)}, & \text{for, } a_R \leq x < b_R \\ 1, & \text{for, } b_R \leq x \end{cases}$$

where a_L, a_R = left and right-hand input limits of trapezoidal dead-band ($M = 0$)
 b_L, b_R = left and right-hand input limits where trapezoid first reaches its maximum value ($M = 1$)

In this definition of the trapezoidal Membership Function, the left-hand limits are associated with a negative slope, while those for the positive limits are associated with a positive slope.

Now if the probability density function of x, $f_x(x)$, is Gaussian:

$$f_x(\hat{x}) = \frac{1}{\sqrt{2\pi}\sigma_x} e^{-\left(\frac{(\hat{x}-\mu_x)^2}{2\sigma_x^2}\right)}$$

where, \hat{x} = estimate of x
 σ_x = standard deviation of \hat{x}
 μ_x = mean of \hat{x}

The corresponding Proposition mean Membership Function, μ_M , is obtained as follows:

$$\mu_M = \frac{1}{\sqrt{2\pi}\sigma_x} \int_{-\infty}^{b_L} e^{-\frac{(\hat{x}-\mu_x)^2}{2\sigma_x^2}} d\hat{x} + \frac{1}{(b_L - a_L)\sqrt{2\pi}\sigma_x} \int_{b_L}^{a_L} (\hat{x} - a_L) e^{-\frac{(\hat{x}-\mu_x)^2}{2\sigma_x^2}} d\hat{x} \\ + \frac{1}{(b_R - a_R)\sqrt{2\pi}\sigma_x} \int_{a_R}^{b_R} (\hat{x} - a_R) e^{-\frac{(\hat{x}-\mu_x)^2}{2\sigma_x^2}} d\hat{x} + \frac{1}{\sqrt{2\pi}\sigma_x} \int_{b_R}^{\infty} e^{-\frac{(\hat{x}-\mu_x)^2}{2\sigma_x^2}} d\hat{x}$$

The general Output Function, $O(x)$, based on the input x is:

$$O(x) = \begin{cases} -c, & \text{for, } x < a_L \\ 0, & \text{for, } a_L \leq x < a_R \\ c, & \text{for, } a_R \leq x \end{cases}$$

where c = consequent discrete output value (e.g.: 7.5, 15, 22.5, 30, 37.5, or 45)

The Proposition mean Output, μ_O , is then:

$$\mu_O = \frac{-c}{\sqrt{2\pi}\sigma_x} \int_{-\infty}^{a_L} e^{-\left(\frac{(\hat{x}-\mu_x)^2}{2\sigma_x^2}\right)} d\hat{x} + \frac{c}{\sqrt{2\pi}\sigma_x} \int_{a_R}^{\infty} e^{-\left(\frac{(\hat{x}-\mu_x)^2}{2\sigma_x^2}\right)} d\hat{x}$$

The general Weight, $W(x)$, is given by:

$$W(x) = \begin{cases} 5, & \text{for, } x < b_L \\ -\frac{5(x - a_L)(x + a_L - 2b_L)}{(b_L - a_L)^2}, & \text{for, } b_L \leq x < a_L \\ 0, & \text{for, } a_L \leq x < a_R \\ -\frac{5(x - a_R)(x + a_R - 2b_R)}{(b_R - a_R)^2}, & \text{for, } a_R \leq x < b_R \\ 5, & \text{for, } b_R \leq x \end{cases}$$

The corresponding Proposition mean Weight, μ_w , is obtained as follows:

$$\begin{aligned} \mu_w = & \frac{5}{\sqrt{2\pi}\sigma_x} \int_{-\infty}^{b_L} e^{-\frac{(\hat{x}-\mu_x)^2}{2\sigma_x^2}} d\hat{x} + \frac{5}{\sqrt{2\pi}\sigma_x} \int_{b_R}^{\infty} e^{-\frac{(\hat{x}-\mu_x)^2}{2\sigma_x^2}} d\hat{x} \\ & - \frac{5}{(b_L - a_L)^2 \sqrt{2\pi}\sigma_x} \int_{b_L}^{a_L} (\hat{x} - a_L)(\hat{x} + a_L - 2b_L) e^{-\frac{(\hat{x}-\mu_x)^2}{2\sigma_x^2}} d\hat{x} \\ & - \frac{5}{(b_R - a_R)^2 \sqrt{2\pi}\sigma_x} \int_{a_R}^{b_R} (\hat{x} - a_R)(\hat{x} + a_R - 2b_R) e^{-\frac{(\hat{x}-\mu_x)^2}{2\sigma_x^2}} d\hat{x} \end{aligned}$$

Then the general Firing Strength, $S(x)$,:

$$S(x) = \begin{cases} -5c, & \text{for, } x < b_L \\ \frac{5c(x - a_L)(x + a_L - 2b_L)}{(b_L - a_L)^2}, & \text{for, } b_L \leq x < a_L \\ 0, & \text{for, } a_L \leq x < a_R \\ -\frac{5c(x - a_R)(x + a_R - 2b_R)}{(b_R - a_R)^2}, & \text{for, } a_R \leq x < b_R \\ 5c, & \text{for, } b_R \leq x \end{cases}$$

The corresponding Proposition mean Firing Strength, μ_s , is obtained as follows:

$$\begin{aligned} \mu_s = & -\frac{5c}{\sqrt{2\pi}\sigma_x} \int_{-\infty}^{b_L} e^{-\frac{(\hat{x}-\mu_x)^2}{2\sigma_x^2}} d\hat{x} + \frac{5c}{\sqrt{2\pi}\sigma_x} \int_{b_R}^{\infty} e^{-\frac{(\hat{x}-\mu_x)^2}{2\sigma_x^2}} d\hat{x} \\ & + \frac{5c}{(b_L - a_L)^2 \sqrt{2\pi}\sigma_x} \int_{b_L}^{a_L} (\hat{x} - a_L)(\hat{x} + a_L - 2b_L) e^{-\frac{(\hat{x}-\mu_x)^2}{2\sigma_x^2}} d\hat{x} \\ & - \frac{5c}{(b_R - a_R)^2 \sqrt{2\pi}\sigma_x} \int_{a_R}^{b_R} (\hat{x} - a_R)(\hat{x} + a_R - 2b_R) e^{-\frac{(\hat{x}-\mu_x)^2}{2\sigma_x^2}} d\hat{x} \end{aligned}$$

The integrals involving integration to $+\infty$ OR $-\infty$ for the mean Weight and mean Firing Strength can be evaluated using the complementary error function, erfc , which is readily available in MATLAB. The remaining integrals have to be evaluated numerically. Since these latter integrals involve finite integration limits, this can easily be performed in MATLAB.

Comparing the expression for mean output with the figure of merit, derived in Section 4.2, it can be seen that the sum of the products of the original figures of merit times their corresponding Proposition output is the mean Proposition output. By comparing the actual output to the mean output of the estimate, the mean output error is obtained. Stated in another way, rather than enumerating the individual Proposition decision error cases, the mean Proposition output provides a direct measure of the decision error when this mean output is compared to the actual output.

4.3 Relative Ground Speed

For the Relative Ground Speed Proposition of Figure 4-1, the parameters required in the above equations are summarized in Table 4-3.

Table 4-3 Relative Ground Speed Proposition Parameters

| Membership Function | b_L | a_L | a_R | b_R | Consequent Function | c |
|---------------------|-------|-------|-------|-------|----------------------|-----|
| 'Is Faster' | -60 | -20 | 20 | 60 | 'Marginally Favored' | 7.5 |

In addition, the probability density for the Relative Ground Speed is:

$$f_{\Delta V_G}(\Delta \hat{V}_G) = \frac{1}{\sqrt{2\pi}\sigma_{\Delta V_G}} e^{-\left(\frac{(\Delta \hat{V}_G - \mu_{\Delta V_G})^2}{2\sigma_{\Delta V_G}^2}\right)}$$

$$\text{where, } \sigma_{\Delta V_G} \equiv \sqrt{\sigma_{V_{G,A}}^2 + \sigma_{V_{G,B}}^2}$$

$$\mu_{\Delta V_G} \equiv (\mu_{V_{G,A}} - \mu_{V_{G,B}})$$

With definition of the Relative Ground Speed Proposition parameters and statistics, the equations of the Section 4.2 can be used to determine the Relative Ground Speed mean Membership, Output, Weight, and Firing Strength.

4.4 Relative ETA Magnitude

Since the Relative ETA Magnitude Proposition input involves a magnitude operation, the probability density function is more complicated than for the other input variables. Hence, the new figures of merit for this variable are derived separately.

The membership, $M(\tau)$ for the Relative ETA Magnitude, τ , is:

$$M(\tau) = \begin{cases} \frac{(60 - \tau)}{60}, & \text{for, } 0 \leq \tau < 60 \text{ sec} \\ 0, & \text{for, } \tau \geq 60 \text{ sec} \end{cases}$$

The Output, $O(\tau)$ is:

$$O(\tau) = \begin{cases} 7.5, & \text{for, } 0 \leq \tau < 60 \text{ sec} \\ 0, & \text{for, } 60 \text{ sec} \leq \tau \end{cases}$$

The Weight, $W(\tau)$ is:

$$W(\tau) = \begin{cases} \frac{(60 - \tau)(60 + \tau)}{720}, & \text{for } 0 \leq \tau < 60 \\ 0, & \text{for } \tau \geq 60 \end{cases}$$

The Firing Strength, $S(\tau)$, is:

$$S(\tau) = \begin{cases} \frac{(60-\tau)(60+\tau)}{96}, & \text{for } 0 \leq \tau < 60 \\ 0, & \text{for } \tau \geq 60 \end{cases}$$

Finally, the probability density function, $f_{\tau}(\tau)$, for the Relative ETA Magnitude (Mueller, 1998) is:

$$f_{\tau}(\hat{\tau}) = \frac{e^{-\frac{(\hat{\tau}-\mu_{\tau})^2}{2\sigma_{\tau}^2}} + e^{-\frac{(\hat{\tau}+\mu_{\tau})^2}{2\sigma_{\tau}^2}}}{\sqrt{2\pi}\sigma_{\tau}}, \quad \hat{\tau} \geq 0$$

where, $\hat{\tau}$ = estimate of the Relative ETA Magnitude, τ

$\sigma_{\tau} = \sqrt{\sigma_{ETA_A}^2 + \sigma_{ETA_B}^2}$, standard deviation of the Relative ETA Magnitude

$\mu_{\tau} = (\mu_{ETA_B} - \mu_{ETA_A})$, mean of the Relative ETA Magnitude

The Mean Membership, μ_M , is:

$$\mu_M = \frac{1}{60\sqrt{2\pi}\sigma_{\tau}} \int_0^{60} (60-\hat{\tau}) \left[e^{-\frac{(\hat{\tau}-\mu_{\tau})^2}{2\sigma_{\tau}^2}} + e^{-\frac{(\hat{\tau}+\mu_{\tau})^2}{2\sigma_{\tau}^2}} \right] d\hat{\tau}$$

The Mean Output, μ_O , can now be derived as follows:

$$\mu_O = \frac{7.5}{\sqrt{2\pi}\sigma_{\tau}} \int_0^{60} \left[e^{-\frac{(\hat{\tau}-\mu_{\tau})^2}{2\sigma_{\tau}^2}} + e^{-\frac{(\hat{\tau}+\mu_{\tau})^2}{2\sigma_{\tau}^2}} \right] d\hat{\tau}$$

The Mean Weight, μ_W , is:

$$\mu_W = \frac{1}{720\sqrt{2\pi}\sigma_{\tau}} \int_0^{60} (60-\hat{\tau})(60+\hat{\tau}) \left[e^{-\frac{(\hat{\tau}-\mu_{\tau})^2}{2\sigma_{\tau}^2}} + e^{-\frac{(\hat{\tau}+\mu_{\tau})^2}{2\sigma_{\tau}^2}} \right] d\hat{\tau}$$

The Mean Firing Strength, μ_S , is:

$$\mu_s = \frac{1}{96\sqrt{2\pi}\sigma_\tau} \int_0^{60} (60 - \hat{\tau})(60 + \hat{\tau}) \left[e^{-\frac{(\hat{\tau} - \mu_\tau)^2}{2\sigma_\tau^2}} + e^{-\frac{(\hat{\tau} + \mu_\tau)^2}{2\sigma_\tau^2}} \right] d\hat{\tau}$$

4.5 NSD_{FCTS}

For the three NSD_{FCTS} Propositions, the parameters are summarized in Table 4-4:

Table 4-4 NSD_{FCTS} Proposition Parameters

| Membership Function | b _L | a _L | a _R | b _R | Consequent Function | c |
|-------------------------------|----------------|----------------|----------------|----------------|-------------------------|----|
| 'Significantly Ahead at FCTS' | -4 | -2 | 2 | 4 | 'Significantly Favored' | 45 |
| 'Ahead at FCTS' | -2.5 | -0.5 | 0.5 | 2.5 | 'Favored' | 30 |
| 'Slightly Ahead at FCTS' | -1 | 0 | 0 | 1 | 'Slightly Favored' | 15 |

The probability density function, $f_\eta(\eta)$, for NSD_{FCTS} is:

$$f_\eta(\hat{\eta}) = \begin{cases} \frac{1}{\sqrt{2\pi}\sigma_{\eta_{AB}}} e^{-\frac{(\hat{\eta}_{AB} - \mu_{\eta_{AB}})^2}{2\sigma_{\eta_{AB}}^2}}, & \text{if A ahead of B at FCTS} \\ \frac{1}{\sqrt{2\pi}\sigma_{\eta_{BA}}} e^{-\frac{(\hat{\eta}_{BA} - \mu_{\eta_{BA}})^2}{2\sigma_{\eta_{BA}}^2}}, & \text{if B ahead of A at FCTS} \end{cases}$$

where, $\sigma_{\eta_{AB}} \equiv \sqrt{\sigma_{N_{A,AB}}^2 + \sigma_{N_{B,AB}}^2}$

$$\mu_{\eta_{AB}} \equiv (\mu_{N_{A,AB}} - \mu_{N_{B,AB}})$$

$$\sigma_{\eta_{BA}} \equiv \sqrt{\sigma_{N_{A,BA}}^2 + \sigma_{N_{B,BA}}^2}$$

$$\mu_{\eta_{BA}} \equiv (\mu_{N_{A,BA}} - \mu_{N_{B,BA}})$$

$$\begin{aligned}
\hat{\eta}_{AB} &\equiv (\hat{N}_{A,AB} - \hat{N}_{B,AB}) \\
\hat{\eta}_{BA} &\equiv (\hat{N}_{A,BA} - \hat{N}_{B,BA}) \\
\hat{N}_{A,AB} &\equiv \left(\frac{N}{\Delta d_{AB}} \right) \hat{V}_{G,A}(t_k), \text{ if A ahead of B at FCTS} \\
\hat{N}_{A,BA} &\equiv \left(\frac{N}{\Delta d_{BA}} \right) \hat{V}_{G,A}(t_k), \text{ if B ahead of A at FCTS} \\
\hat{N}_{B,AB} &\equiv \left(\frac{N}{\Delta d_{AB}} \right) \hat{V}_{G,B}(t_k), \text{ if A ahead of B at FCTS} \\
\hat{N}_{B,BA} &\equiv \left(\frac{N}{\Delta d_{BA}} \right) \hat{V}_{G,B}(t_k), \text{ if B ahead of A at FCTS} \\
N &\equiv (t_{FCTS} - t_k) - l_{k,FCTS,A} \left(V_{G,A}(t_{FCTS}) - V_{G,B}(t_{FCTS}) \right)
\end{aligned}$$

The problem with formulating the figures of merit using the generic equations presented in Section 4.3 is that the estimate of NSD_{FCTS} may vary between positive and negative values prior to reaching t_{FCTS} . This depends on whether the aircraft separation distance estimate between aircraft A and B, is positive or negative. What complicates the computation of the estimate of NSD_{FCTS} is that whenever the sign of the estimated separation distance at FCTS switches, the required minimum separation distance will also switch from Δd_{AB} to Δd_{BA} .

A practical formulation involves using the estimate of the SD_{FCTS} , λ , in place of the NSD_{FCTS} . The new figures of merit can be reformulated using SD_{FCTS} and still obtain the same results as would be obtained with the figures of merit for NSD_{FCTS} . The corresponding SD_{FCTS} Proposition parameters are summarized in Table 4-5:

Table 4-5 SD_{FCTS} Proposition Parameters

| Membership Function | b_L | a_L | a_R | b_R | Consequent Function | c |
|-------------------------------|----------------------|----------------------|---------------------|---------------------|-------------------------|----|
| 'Significantly Ahead at FCTS' | $-4 \Delta d_{BA}$ | $-2 \Delta d_{BA}$ | $2 \Delta d_{AB}$ | $4 \Delta d_{AB}$ | 'Significantly Favored' | 45 |
| 'Ahead at FCTS' | $-2.5 \Delta d_{BA}$ | $-0.5 \Delta d_{BA}$ | $0.5 \Delta d_{AB}$ | $2.5 \Delta d_{AB}$ | 'Favored' | 30 |
| 'Slightly Ahead at FCTS' | $-\Delta d_{BA}$ | 0 | 0 | Δd_{AB} | 'Slightly Favored' | 15 |

The probability density function, $f_{\lambda}(\lambda)$, for λ is:

$$f_{\lambda}(\hat{\lambda}) = \frac{1}{\sqrt{2\pi}\sigma_{\lambda}} e^{-\frac{(\hat{\lambda}-\mu_{\lambda})^2}{2\sigma_{\lambda}^2}}$$

where, $\sigma_{\lambda} = \sqrt{\left(N_A \sigma_{V_{G,A}}\right)^2 + \left(N_B \sigma_{V_{G,B}}\right)^2}$
 $\mu_{\lambda} = \left(N_A \mu_{V_{G,A}} - N_B \mu_{V_{G,B}}\right)$
 $N_A = \left(t_{FCTS} - t_k\right) - I_{k,FCTS,A} \left(V_{G,A}(t_{FCTS}) - V_{G,B}(t_{FCTS})\right)$
 $N_B = \left(t_{FCTS} - t_k\right) - I_{k,FCTS,B} \left(V_{G,A}(t_{FCTS}) - V_{G,B}(t_{FCTS})\right)$

The generic figure of merit equations of Section 4.3 can now be used with the above definition of the parameters and the SD statistics.

4.6 Excess Delay

The parameters for the three Propositions for the Excess Delay were presented in Tables 2-10 and 2-11 and are again summarized in Tables 4-6 and 4-7.

Table 4-6 Excess Delay Proposition Parameters (Aircraft A Sequenced Behind B)

| Membership Function | b_L | a_L | a_R | b_R | Consequent Function | c |
|------------------------------|-------|-------|-------|-------|---------------------------|-----|
| 'A is Significantly Delayed' | 0 | 0 | 50 | 100 | 'A is Favored' | 30 |
| 'A is Delayed' | 0 | 0 | 25 | 75 | 'A is Slightly Favored' | 15 |
| 'A is Slightly Delayed' | 0 | 0 | 0 | 50 | 'A is Marginally Favored' | 7.5 |

Table 4-7 Excess Delay Proposition Parameters (Aircraft B Sequenced Behind A)

| Membership Function | b_L | a_L | a_R | b_R | Consequent Function | c |
|------------------------------|-------|-------|-------|-------|---------------------------|------|
| 'B is Significantly Delayed' | -100 | -50 | 0 | 0 | 'B is Favored' | -30 |
| 'B is Delayed' | -75 | -25 | 0 | 0 | 'B is Slightly Favored' | -15 |
| 'B is Slightly Delayed' | -50 | 0 | 0 | 0 | 'B is Marginally Favored' | -7.5 |

The probability density function, $f_{\xi}(\xi)$, for the Excess Delay, ξ , is:

and,

$$f_{\zeta_{BA}}(\hat{\zeta}_{BA}) = \frac{1}{\sqrt{2\pi}\sigma_{\zeta_{BA}}} e^{-\left(\frac{(\hat{\zeta}_{BA} - \mu_{\zeta_{BA}})^2}{2\sigma_{\zeta_{BA}}^2}\right)}, \quad \text{for A sequenced behind B}$$

$$f_{\zeta_{AB}}(\hat{\zeta}_{AB}) = \frac{1}{\sqrt{2\pi}\sigma_{\zeta_{AB}}} e^{-\left(\frac{(\hat{\zeta}_{AB} - \mu_{\zeta_{AB}})^2}{2\sigma_{\zeta_{AB}}^2}\right)}, \quad \text{for B sequenced behind A}$$

where,

$$\sigma_{\zeta_{AB}} = \sqrt{(E_{B,Early})^2 \sigma_{V_{G,B,Early}}^2 + (E_{B,Late})^2 \sigma_{V_{G,B,Late}}^2 + (E_B)^2 \sigma_{V_{G,B}}^2}$$

$$\mu_{\zeta_{AB}} = (E_{B,Early}) \mu_{V_{G,B,Early}} - (E_{B,Late}) \mu_{V_{G,B,Late}} - (E_B) \mu_{V_{G,B}}$$

$$\sigma_{\zeta_{BA}} = \sqrt{(E_{A,Early})^2 \sigma_{V_{G,A,Early}}^2 + (E_{A,Late})^2 \sigma_{V_{G,A,Late}}^2 + (E_A)^2 \sigma_{V_{G,A}}^2}$$

$$\mu_{\zeta_{BA}} = (E_{A,Early}) \mu_{V_{G,A,Early}} - (E_{A,Late}) \mu_{V_{G,A,Late}} - (E_A) \mu_{V_{G,A}}$$

with,

$$E_{B,Early} \equiv D_B (TA_{B,Late} - STA_{B,AB}) I_{TA,Early,B}$$

$$E_{B,Late} \equiv D_B (TA_{B,Early} - STA_{B,AB}) I_{TA,Late,B}$$

$$E_B \equiv D_B (TA_{B,Late} - TA_{B,Early}) I_{STA,B}$$

$$E_{A,Early} \equiv D_A (TA_{A,Late} - STA_{A,BA}) I_{TA,Early,A}$$

$$E_{A,Late} \equiv D_A (TA_{A,Early} - STA_{A,BA}) I_{TA,Late,A}$$

$$E_A \equiv D_A (TA_{A,Late} - TA_{A,Early}) I_{STA,A}$$

$$D_B \equiv \frac{-100}{(TA_{B,Late} - TA_{B,Early})^2}$$

$$D_A \equiv \frac{100}{(TA_{A,Late} - TA_{A,Early})^2}$$

and,

$$I_{STA} \equiv - \left[\int_{t_k}^{STA} \frac{dt}{V_G(t)} \right]$$

$$I_{TA,Early} \equiv - \left[\int_{t_k}^{TA_{Early}} \frac{dt}{V_{G,Early}(t)} \right]$$

and,
$$I_{TA,Late} \equiv - \left[\int_{t_k}^{TA_{Late}} \frac{dt}{V_{G,Late}(t)} \right]$$

4.7 Normalized Delay Savings

The parameters which are required to compute the Proposition statistics for the NDS are summarized in Table 4-8.

Table 4-8. NDS Proposition Parameters

| Membership Function | b_L | a_L | a_R | b_R | Consequent Function | c |
|----------------------------------|-------|-------|-------|-------|---------------------------|------|
| 'A Ahead of B Causes Less Delay' | 0 | 0 | 0 | 2 | 'A is Marginally Favored' | 7.5 |
| 'B Ahead of A Causes Less Delay' | -2 | 0 | 0 | 0 | 'B is Marginally Favored' | -7.5 |

If η is the NDS input, the NDS probability density function, $f_\eta(\eta)$, is defined as follows:

$$f_\eta(\hat{\eta}) = \frac{1}{\sqrt{2\pi}\sigma_\eta} e^{-\frac{(\hat{\eta} - \mu_\eta)^2}{2\sigma_\eta^2}}$$

where,

$$\sigma_\eta \equiv \left[A_1^2 \sigma_{V_{G,A,AB}}^2 + A_2^2 \sigma_{V_{G,A,BA}}^2 + A_3^2 \sigma_{V_{G,B,AB}}^2 + A_4^2 \sigma_{V_{G,B,BA}}^2 + A_5^2 \sigma_{V_{G,A,Early}}^2 + A_6^2 \sigma_{V_{G,B,Early}}^2 \right]^{0.5}$$

$$\mu_\eta \equiv A_1 \mu_{V_{G,A,AB}} - A_2 \mu_{V_{G,A,BA}} + A_3 \mu_{V_{G,B,AB}} - A_4 \mu_{V_{G,B,BA}} - A_5 \mu_{V_{G,A,Early}} - A_6 \mu_{V_{G,B,Early}}$$

with, $A_1 \equiv \left(\frac{l(t_k)_{STA,A,AB}}{\Delta t_{AB}} \right)$, $A_2 \equiv \left(\frac{l(t_k)_{STA,A,BA}}{\Delta t_{BA}} \right)$, $A_3 \equiv \left(\frac{l(t_k)_{STA,B,AB}}{\Delta t_{AB}} \right)$

$$A_4 \equiv \left(\frac{l(t_k)_{STA,B,BA}}{\Delta t_{BA}} \right), \quad A_5 \equiv l(t_k)_{TA,Early,A} \left(\frac{1}{\Delta t_{AB}} - \frac{1}{\Delta t_{BA}} \right)$$

$$A_6 \equiv l(t_k)_{TA,Early,B} \left(\frac{1}{\Delta t_{AB}} - \frac{1}{\Delta t_{BA}} \right)$$

$$I(t_k)_{STA} \equiv - \left[\int_{t_k}^{STA} \frac{dt}{V_G(t)} \right]$$

$$I(t_k)_{TA,Early} \equiv - \left[\int_{t_k}^{TA_{Early}} \frac{dt}{V_{G,Early}(t)} \right]$$

4.8 Controllability

The input parameters for the Controllability Propositions are summarized in Table 4-9.

Table 4-9. Controllability Proposition Parameters

| Membership Function | b_L | a_L | a_R | b_R | Consequent Function | c |
|---|-------|-------|-------|-------|---------------------------|-----|
| 'Significantly out of Delay' (Aircraft A behind B) | -0.5 | 0.25 | 0 | 0 | 'A Significantly Favored' | 45 |
| 'Significantly out of Delay' (Aircraft B behind A) | 0 | 0 | -0.25 | 0.5 | 'B Significantly Favored' | -45 |
| 'Out of Delay' (Aircraft A behind B) | 0 | 0.75 | 0 | 0 | 'A Favored' | 30 |
| 'Out of Delay' (Aircraft B behind A) | 0 | 0 | -0.75 | 0 | 'B Favored' | -30 |

The probability density function for the Controllability, κ , is:

and,

$$f_{\kappa_{BA}}(\hat{\kappa}_{BA}) = \frac{1}{\sqrt{2\pi}\sigma_{\kappa_{BA}}} e^{-\left(\frac{(\hat{\kappa}_{BA} - \mu_{\kappa_{BA}})^2}{2\sigma_{\kappa_{BA}}^2}\right)}, \quad \text{for A sequenced behind B}$$

$$f_{\kappa_{AB}}(\hat{\kappa}_{AB}) = \frac{1}{\sqrt{2\pi}\sigma_{\kappa_{AB}}} e^{-\left(\frac{(\hat{\kappa}_{AB} - \mu_{\kappa_{AB}})^2}{2\sigma_{\kappa_{AB}}^2}\right)}, \quad \text{for B sequenced behind A}$$

$$\begin{aligned}
\text{where, } \sigma_{\kappa_{AB}} &= \sqrt{\left(\frac{l(t_k)_{STA,B,AB}}{\Delta t_{AB}}\right)^2 \sigma_{V_{G,B}}^2 + \left(\frac{l(t_k)_{TA,Late,B}}{\Delta t_{AB}}\right)^2 \sigma_{V_{G,B,Late}}^2} \\
\mu_{\kappa_{AB}} &= \left(\frac{l(t_k)_{STA,B,AB}}{\Delta t_{AB}}\right) \mu_{V_{G,B}} - \left(\frac{l(t_k)_{TA,Late,B}}{\Delta t_{AB}}\right) \mu_{V_{G,B,Late}} \\
\sigma_{\kappa_{BA}} &= \sqrt{\left(\frac{l(t_k)_{STA,A,BA}}{\Delta t_{BA}}\right)^2 \sigma_{V_{G,A}}^2 + \left(\frac{l(t_k)_{TA,Late,A}}{\Delta t_{BA}}\right)^2 \sigma_{V_{G,A,Late}}^2} \\
\mu_{\kappa_{BA}} &= \left(\frac{l(t_k)_{STA,A,BA}}{\Delta t_{BA}}\right) \mu_{V_{G,A}} - \left(\frac{l(t_k)_{TA,Late,A}}{\Delta t_{BA}}\right) \mu_{V_{G,A,Late}}
\end{aligned}$$

5.0 ORDERING PROCEDURE PERFORMANCE SIMULATION RESULTS

In this Chapter the FAST Scheduling Logic error models developed in the last Chapter are incorporated into a performance simulation for the Ordering Procedure for a GENERAL-Type Spatial Constraint. While the focus is on the Propositions which are directly or indirectly dependent on the Relative Ground Speed errors, it is necessary to also model the remaining Propositions. For these latter Propositions, the inputs are assumed to be error-free with respect to the Relative Ground Speed dependent Propositions. By modeling all the Propositions for each Procedure, it is possible to determine the influence that the Relative Ground Speed errors have on the Procedure decision. A listing of this simulation is presented in Appendix B.

5.1 Ordering Procedure Performance Simulation

Table 5-1 summarizes the seven Proposition pairs which are evaluated by the Ordering Procedure of a GENERAL-Type Spatial Constraint. It specifically indicates the Proposition, the Proposition input, the Consequent, and the Consequent Output. Propositions 2 and 4 are the same since FAST SL uses the same logic for both the Ordering and the Merging Procedure of a GENERAL-Type Spatial Constraint. However, for the latter, the first three Propositions are referenced to the FCTS rather than the current position.

Table 5-1. Ordering Procedure of a GENERAL-Type Spatial Constraint

| Number | Proposition | Input | Consequent | Output |
|--------|--|---------------------------|-------------------------|--------|
| 1 | 'Significantly ahead at current position' | NSD_{TRK} | 'Significantly favored' | +/-45 |
| 2 | 'Ahead at current position' | NSD_{TRK} | 'Favored' | +/-30 |
| 3 | 'Slightly ahead at current position' | NSD_{TRK} | 'Slightly favored' | +/-15 |
| 4 | 'Ahead at current position' | NSD_{TRK} | 'Slightly favored' | +/-15 |
| 5 | 'Faster at current position' | ΔV_G | 'Marginally favored' | +/-7.5 |
| 6 | 'Lower at current position' | Δh | 'Marginally favored' | +/-7.5 |
| 7 | ('Close') AND ('Faster at current position') | Δd & ΔV_G | 'Slightly favored' | +/-15 |

These Propositions are evaluated for two aircraft at a time, such as aircraft A and B. If the Proposition is true for aircraft A ahead of B, then the positive Output is selected. If

the Proposition is true for aircraft B ahead of A, then the negative Output is selected. By combining these individual Proposition Outputs in a Weighted sense (the Firing Strength), the Procedure decision is obtained. Hence if the Procedure decision is positive, aircraft A is recommended to be ahead of aircraft B. If the Procedure decision is negative, aircraft B is recommended to be ahead of aircraft A.

Figure 5-1 summarizes the principal modules of the FAST SL Ordering Procedure Performance Simulation and their relationship. The FAST TS simulation (fastTS.m), which was previously developed and documented in (Mueller, 1998) is used to generate a nominal trajectory for the two aircraft (A and B) whose order is evaluated.

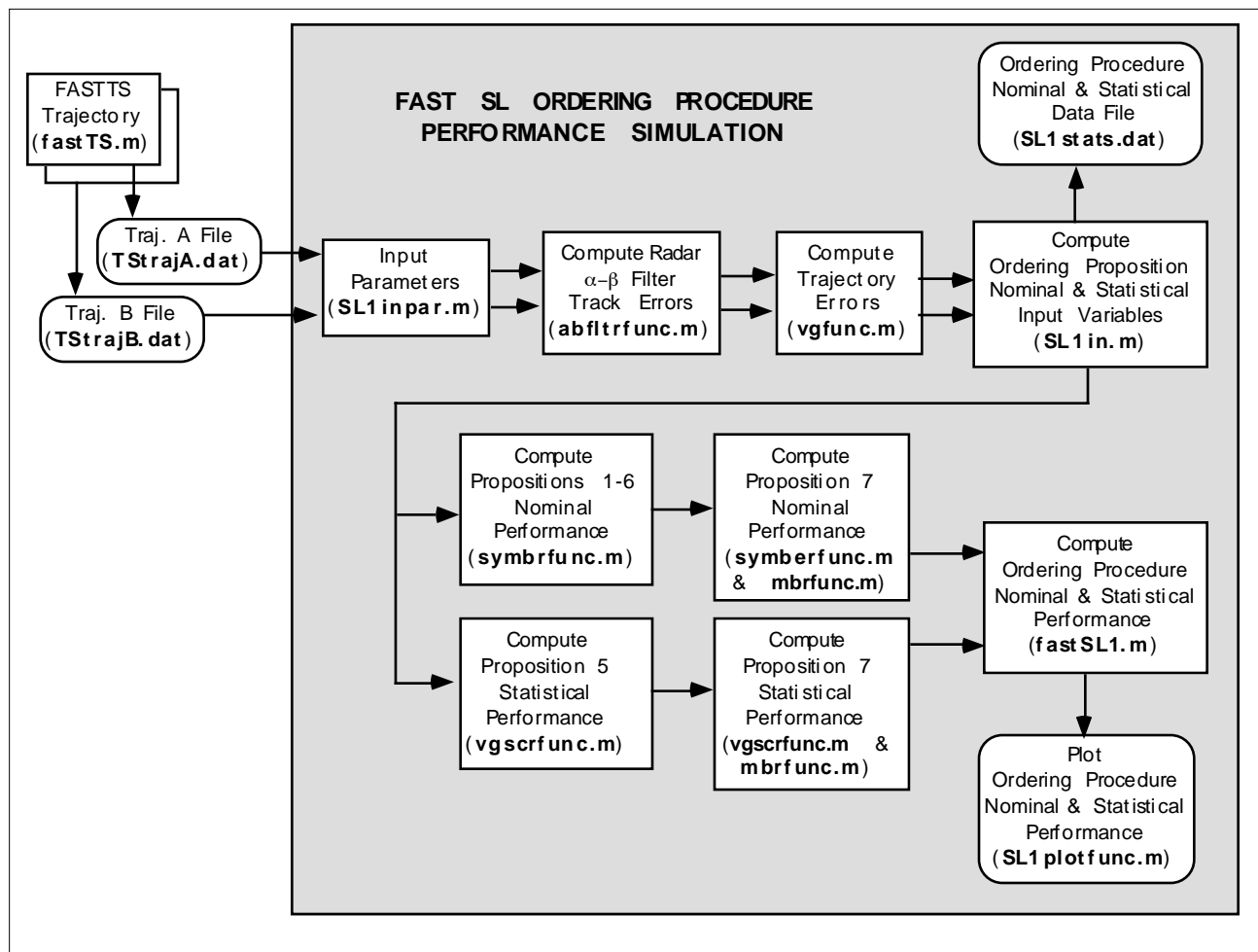


Figure 5-1. FAST SL Ordering Procedure Performance Simulation

The control parameters for this performance simulation, fastSL1.m, are specified in SL1inpar.m. Using these parameters, the nominal trajectories are loaded in and used as an input to the α - β radar tracking filter error simulation (abfltrfunc.m) which was developed and documented in (Mueller, 1998). Next, the trajectory errors are computed for these two trajectories in vgfunc.m. With the nominal trajectories and their error

histories, the aircraft nominal relative trajectory variables and their error statistics are computed in SL1in.m.

The statistical inputs as well as the nominal inputs are then used individually to evaluate the seven Propositions of the Ordering Procedure. As shown in this figure, the Relative Ground Speed statistics, which are inputs to Propositions 5 and 7, are the only ground speed-dependent input used in this Procedure. The combined Firing Strength from these seven Propositions determines the nominal and the statistical Ordering Procedure performance (fastSL1.m). By comparing the performance of this Procedure under nominal as well as perturbed conditions, the degradation due to the Relative Ground Speed errors can be determined.

Under this study, the principal figures of merit which are used are the expected (mean) Proposition Membership, Output, Weight, and Firing Strength. The effects of the corresponding dispersions (standard deviations) are only used to loosely determine a confidence interval, such as the 95% confidence interval corresponding to ± 2 sigma about the estimate, assuming that the statistics are approximately Gaussian.

5.2 Nominal Trajectories

To evaluate the performance of the Ordering Procedure Procedure, trajectories for two aircraft are required. Hence, a nominal jet trajectory from the Southwest Metering Fix to Dallas-Ft. Worth Runway 18R is used. Then, to obtain the trajectories for two jet aircraft in-track from each other, the nominal jet trajectory can be used twice with different time biases as illustrated in Figure 5-2.

For convenience the period of interest is the time interval from the metering fix to the FCTS of the jet aircraft with a turboprop merging onto the common Downwind flight path segment. This merging scenario is evaluated in the next Chapter for the Merging Procedure.

Since the performance degradation due to Relative Ground Speed errors is the principal focus of this study, it is desirable to select the relative spacing such that minimum separation criteria are satisfied at the FCTS under nominal conditions. Under this nominal situation, no air traffic controller intervention is required. When the Relative Ground Speed errors are added, it is possible to determine if an incorrect Procedure decision is produced which might lead to an incorrect air traffic controller intervention. Hence, in Figure 5-2 the two jets are separated by 3 nm from each other at their FCTS with the turboprop.

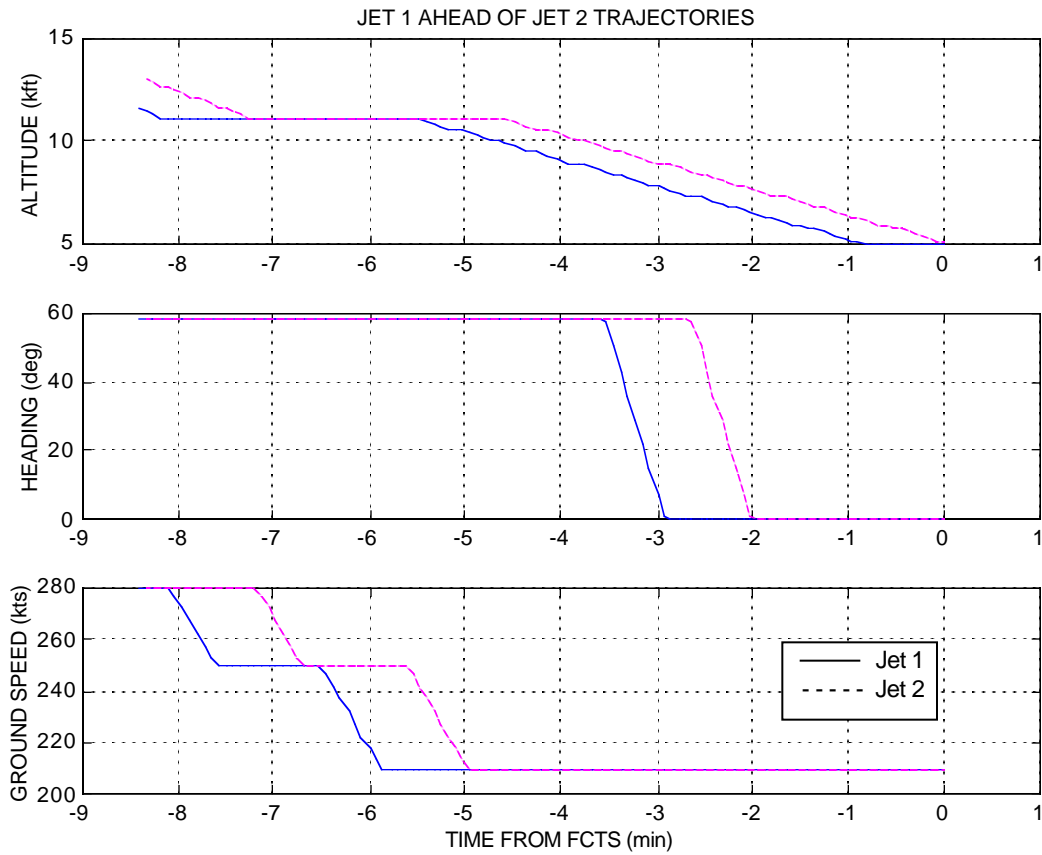


Figure 5-2. FAST TS Jet 1 and Jet 2 Time Histories

5.3 Ordering Procedure Input Error Statistics

In this section, the FAST SL Proposition input error statistics are presented. These estimation errors will be combined, in the performance simulation of the Ordering Procedure, with the nominal (error-free) inputs to obtain the estimated inputs. For the Ordering Procedure, the focus is on the Relative Ground Speed estimation error statistics. These are the only Proposition inputs used by the Ordering Procedure of a GENERAL-Type Spatial Constraint which are dependent on ground speed errors.

The case of the Relative Ground Speed error for Jet 1 ahead of Jet 2 is illustrated in Figure 5-3. The Relative Ground Speed estimation error mean must be added to the nominal Relative Ground Speed to obtain the Relative Ground Speed estimate. The standard deviation of the Relative Ground Speed estimate is the Relative Ground Speed error standard deviation. Also shown in this figure is a 95% confidence (± 2 sigma) bound around the estimate.

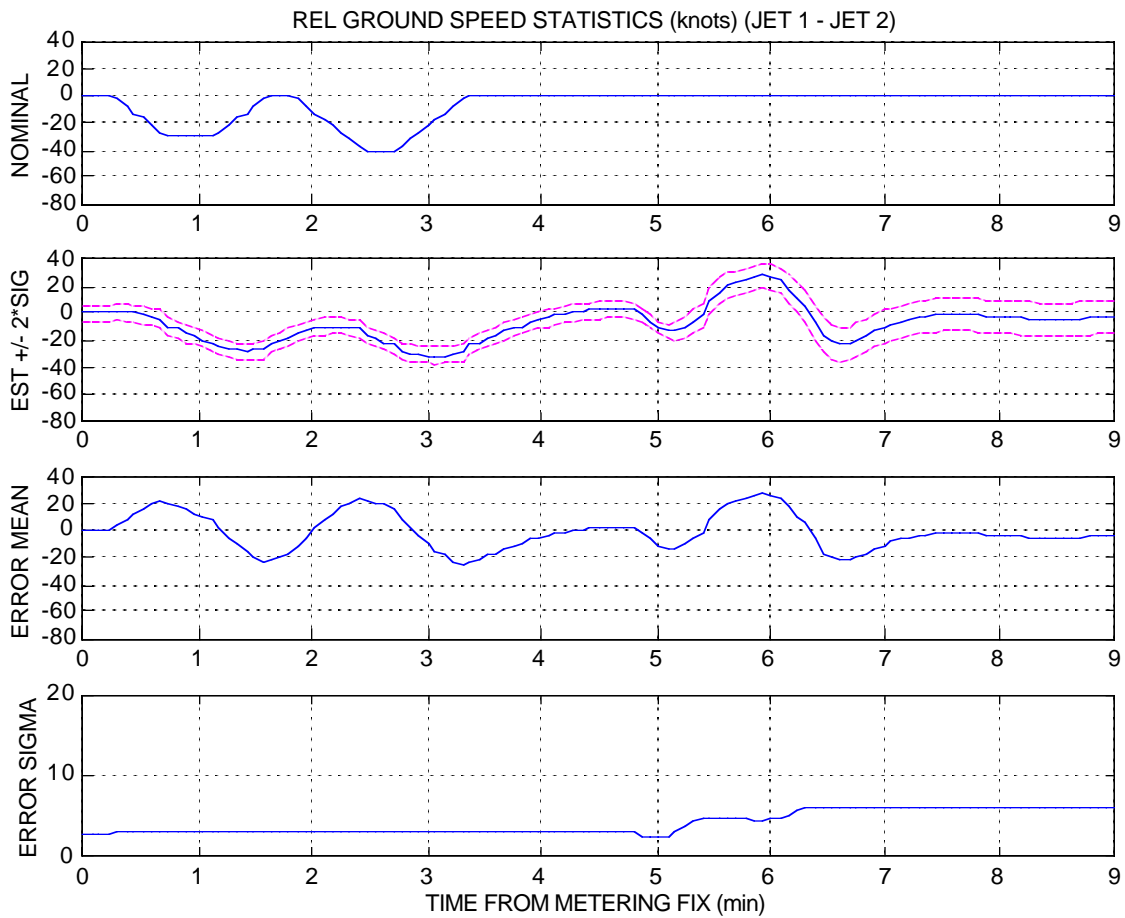


Figure 5-3. Relative Ground Speed Statistics (Jet 1 ahead of Jet 2)

As can be seen in this figure, the estimate 'lags' the nominal Relative Ground Speed history with the addition of the mean error. On the other hand, the standard deviation is so small that it leads to a very tight 95% confidence bound around the estimate.

Examining this figure further, it can be seen that the Relative Ground Speed mean varies between ± 30 kts while the standard deviation is generally less than 6 kts. The individual spikes observed in the mean error histories correspond to either one or the other aircraft undergoing a speed reduction maneuver or a heading change.

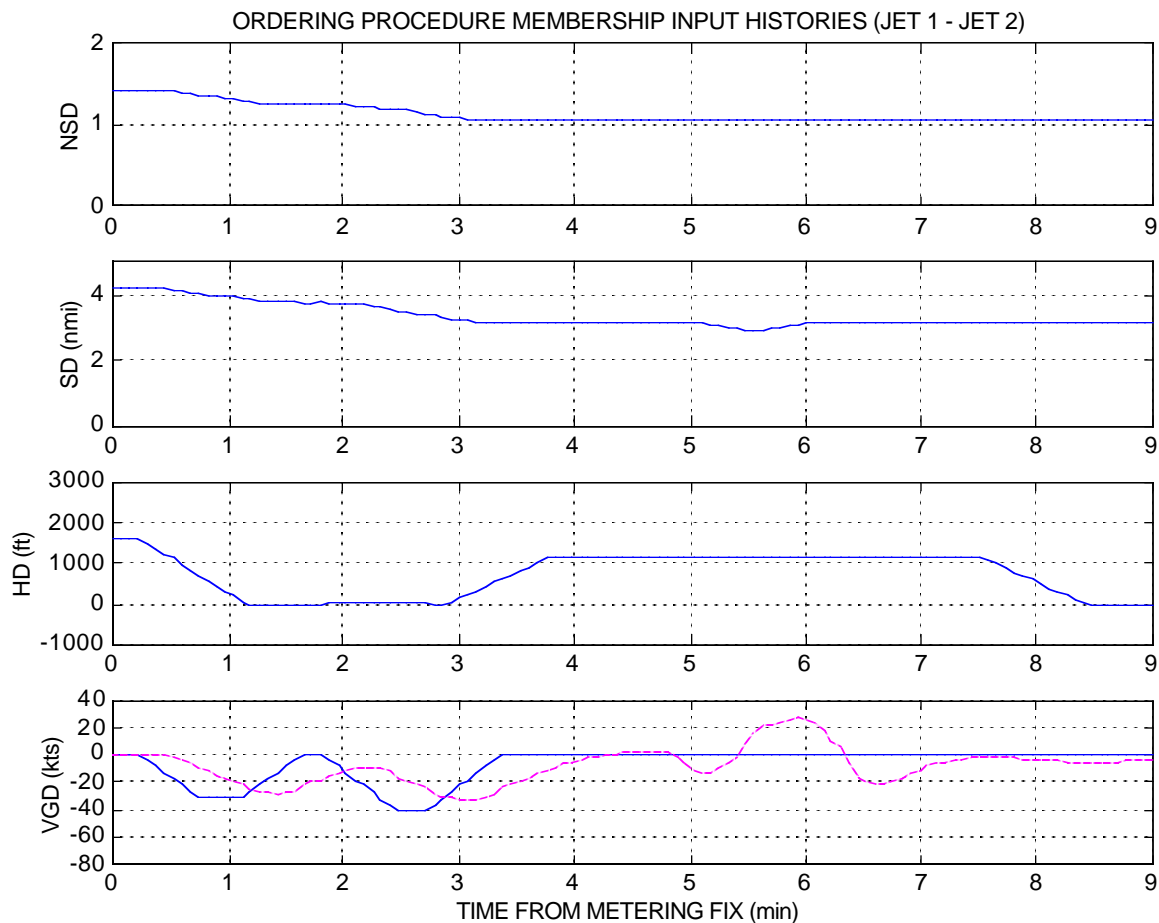
5.4 Ordering Procedure Performance Statistics

In this section the performance of the Ordering Procedure of a GENERAL-Type Spatial Constraint is evaluated. The seven Propositions which are used by this Procedure are summarized in Table 5-1. The performance is determined using the input variables for the Jet 1 followed by a Jet 2 trajectory history.

It is necessary to determine the impact on this Procedure from the decision errors introduced by those Propositions which are sensitive to Relative Ground Speed errors. Since the only ground speed dependent Proposition inputs are the Relative Ground Speed itself, the two Propositions (5 and 7) which use this input are evaluated with both the nominal and the estimated Relative Ground Speed. The remaining Proposition inputs are assumed to be nearly error-free with respect to the Relative Ground Speed errors. Hence, only the nominal Proposition inputs are used for those Propositions.

5.4.1 Membership Statistics

Figure 5-4 presents the input variable histories for the Proposition pairs evaluated by the Ordering Procedure. The solid curves represent the nominal inputs while the dashed curves represent the statistical estimates.



5-4. Ordering Procedure Input Histories

Differences between the nominal Normalized Separation Distance (NSD) and the Horizontal Separation histories arise from the fact that the former is based on path separation distance while the latter is based on line-of-sight separation distance. Hence when Jet 1 reaches the Downwind path segment before Jet 2, the Horizontal Separation

is shorter than the path separation distance. The histories are based on the Jet 1 - Jet 2 relative trajectories for the time history from the metering fix to the time of FCTS.

Also shown in Figure 5-4 is the relationship of the nominal and the estimated Relative Ground Speed. The former is shown in solid while the latter in dashed form. This subplot shows how the estimate lags the nominal when speed reduction or heading maneuvers are made by one or the other aircraft.

Figure 5-5 presents the Membership histories for the seven Propositions of Table 5-1.

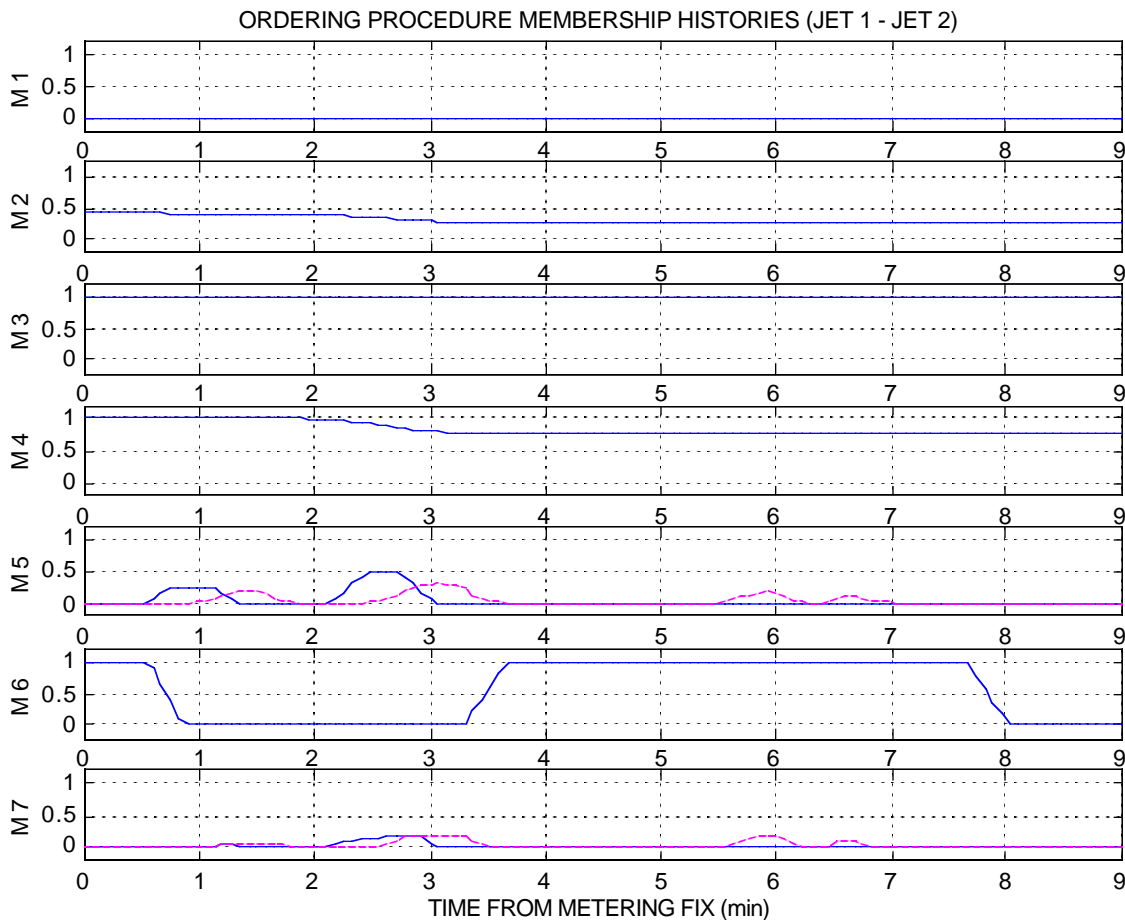


Figure 5-5. Ordering Procedure Membership Histories (Jet 1 ahead of Jet 2)

In this figure the nominal Membership values are shown with a solid line while the estimates are shown with a dashed line. The first four Membership histories are based on the Propositions which use the current Normalized Separation Distance between the two jets as the input.

The fifth Proposition is based on the Relative Ground Speed. This Membership history is seen to vary from zero to one. The seventh Proposition also uses the Relative Ground Speed and the Horizontal Separation as joint inputs. The AND operation selects the input which leads to minimum Membership -- the one which produces the lowest

Membership value. Finally, the sixth Proposition uses the Relative Altitude as input. The Membership for this Proposition varies between zero and one, depending on whether the two jets have the same altitude.

5.4.2 Output Statistics

Figure 5-6 presents the Proposition Output histories. The histories for the nominal Output are shown with a solid line while those for the estimates are shown with a dashed line.

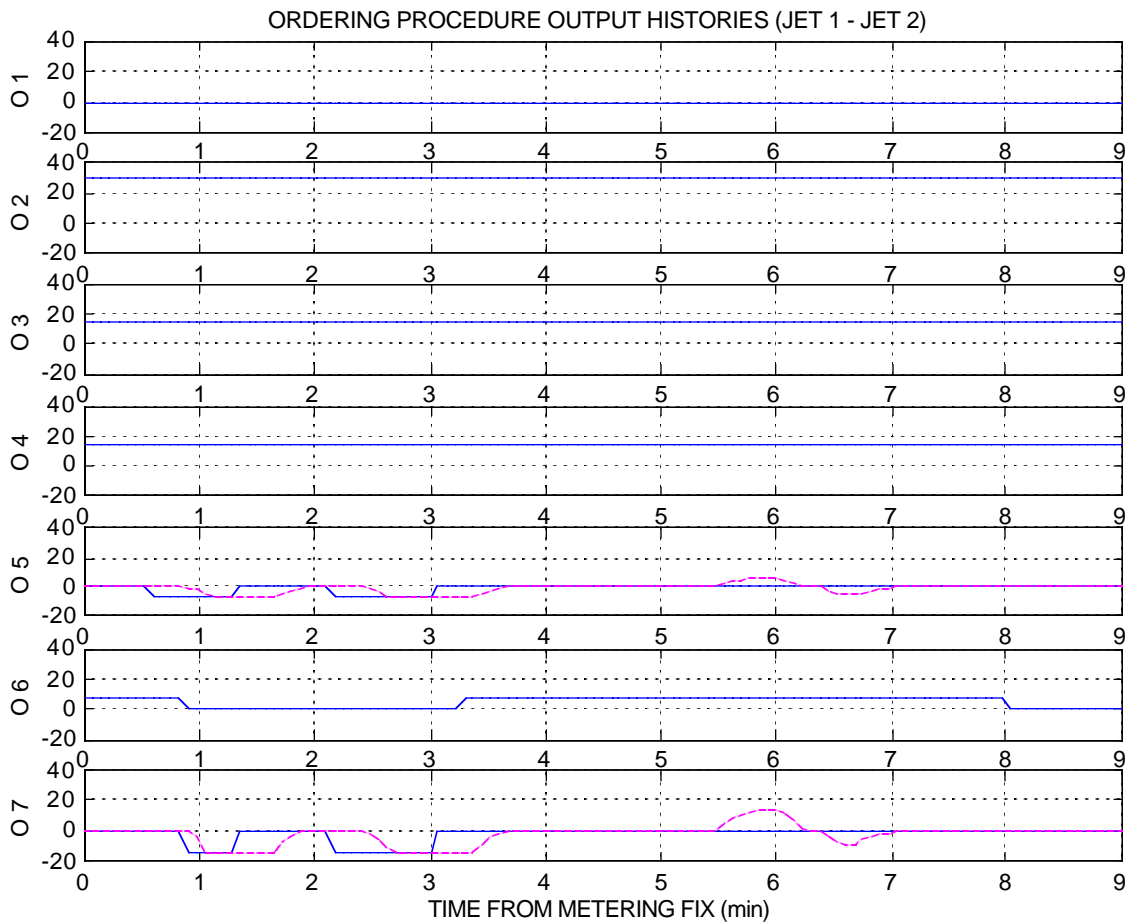


Figure 5-6. Ordering Procedure Output Histories (Jet 1 ahead of Jet 2)

The Output value is nonzero, if the Membership is nonzero. The maximum Output value can reach +/-45, depending on the Proposition. However, while the Membership can take any value between zero and one, the nominal Output can only take the discrete positive or negative value assigned to that Proposition or zero. The Output estimate, however, can take on any value between the maximum nominal positive and negative Output for that Proposition.

Examining the first four Propositions, it can be seen that these all have positive or zero Output values. This corresponds to the fact that Jet 1 always remains ahead of Jet 2. The fifth and seventh Proposition show both positive and negative Output values. This arises from the fact that the Relative Ground Speed may be negative during the early stages of this history since Jet 2 performs its speed reductions after Jet 1. Hence, it appears at times that Jet 2 might overtake Jet 1. The Output value for the sixth Proposition varies between zero and 7.5. This arises from the fact that Jet 2 performs its altitude reduction maneuvers later than Jet 1. This results in the condition where Jet 1 is lower than Jet 2, even though they are both flying along the same flight path history.

5.4.3 Weight Statistics

Figure 5-7 presents the Proposition Weight histories. The Weight histories can only have positive values between 0 and 5.

5.4.4 Firing Strength Statistics

Figure 5-8 presents the Firing Strength histories for these seven Propositions. The Firing Strengths corresponding to the nominal Proposition inputs are shown with a solid line while those corresponding to estimated Proposition inputs are shown with a dashed line. While the Firing Strength can vary between ± 225 , the vertical axes were set at -50 and 150 in this Figure.

This Figure shows that the Firing Strength histories are positive or zero except for the fifth and seventh Proposition. Both of the latter two are dependent on the Relative Ground Speed.

Since these Firing Strength histories are partly determined by the maximum nominal Output for each Proposition, it is convenient to normalize these by the sum of all the Proposition Weights. These Normalized Firing Strength histories are illustrated in Figure 5-9.

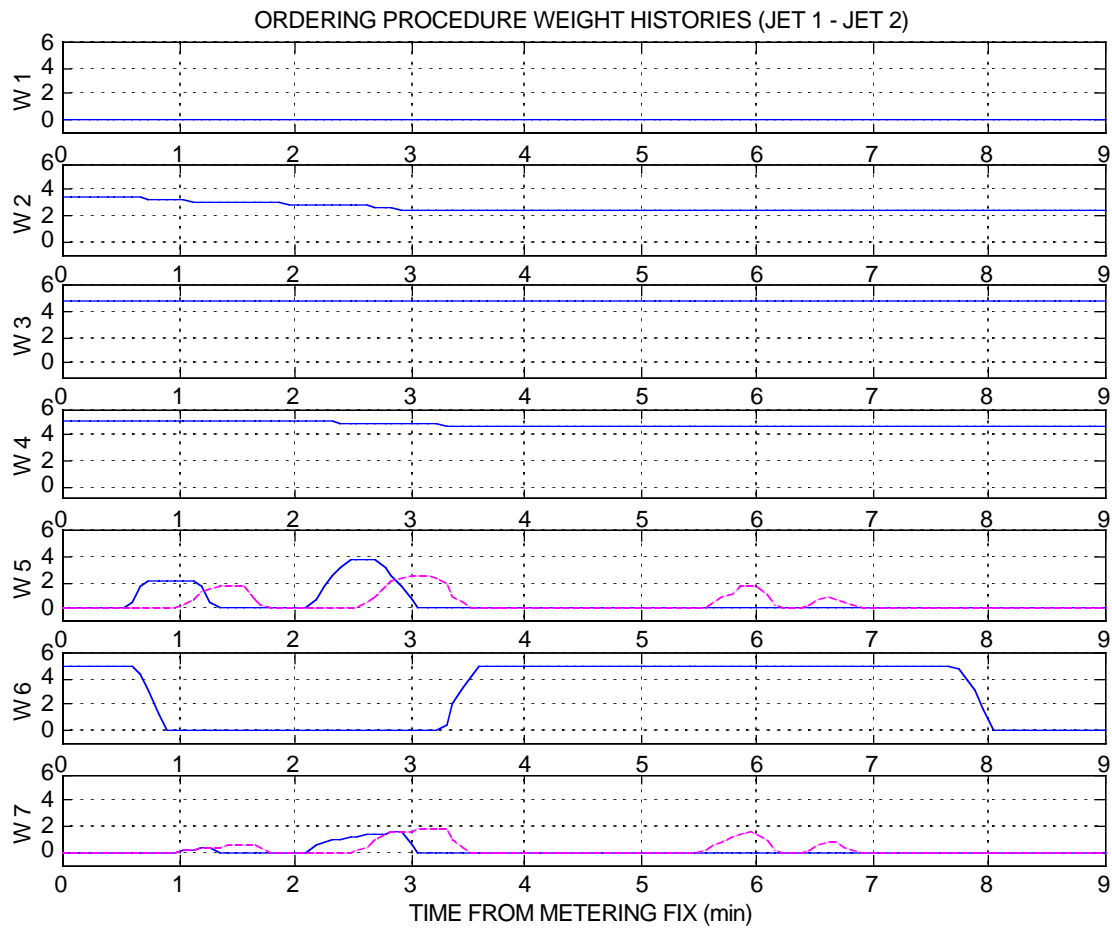


Figure 5-7. Ordering Procedure Weight Histories (Jet 1 ahead of Jet 2)

5.4.5 Proposition 5 Results

A detailed look at the Proposition 5 results is presented in Figure 5-10. This figure shows the nominal results with a solid line. The estimates are shown with a dashed line while the 95% confidence interval about the estimate are shown with dotted lines.

The Output values can vary between ± 7.5 . For the nominal Output, only the discrete Output values of 0 or ± 7.5 are possible. For the estimate as well as its 95% confidence interval, the histories can vary continuously between ± 7.5 .

The Weight is limited to values between 0 and 5. This leads to Firing Strength values bounded by ± 37.5 . The normalized Firing Strength history for this case appears to be bounded by ± 2 .

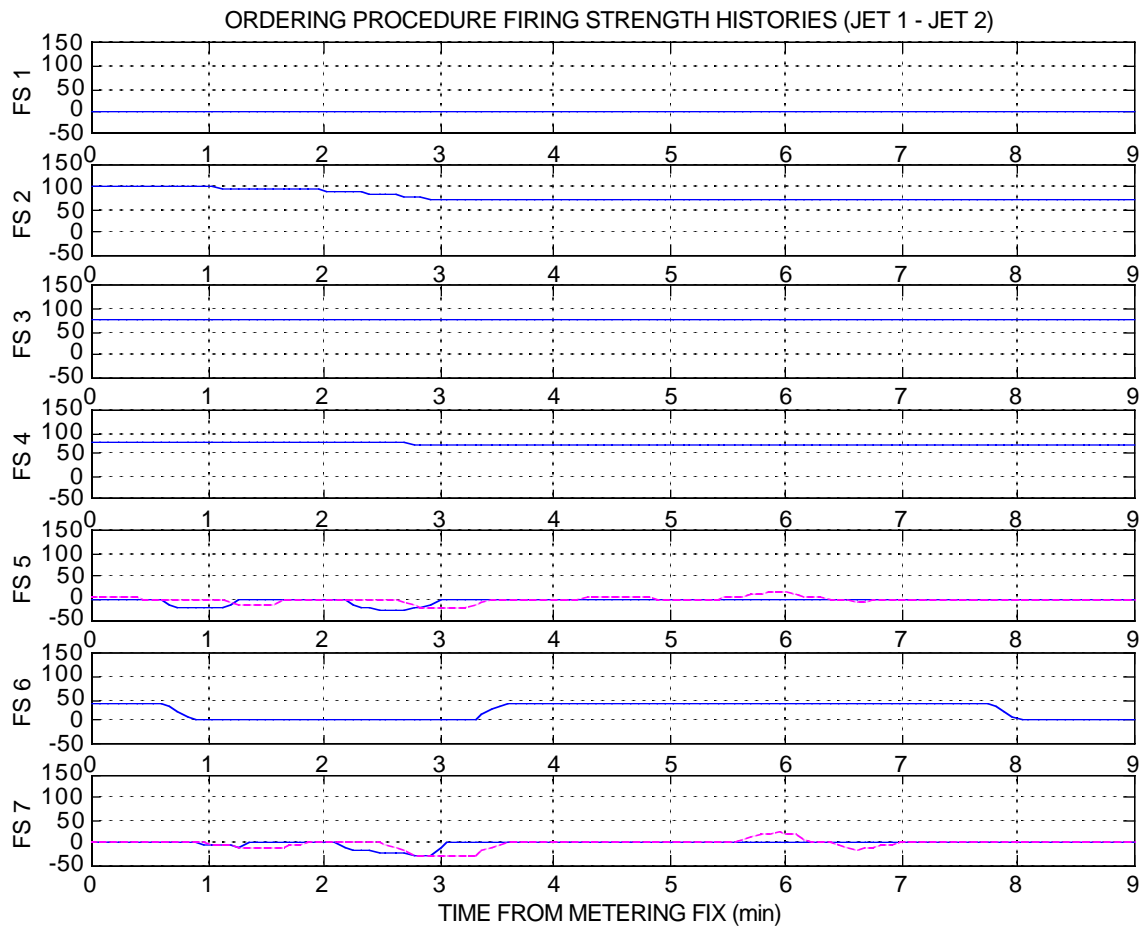


Figure 5-8. Ordering Procedure Firing Strength Histories (Jet 1 ahead of Jet 2)

5.4.6 Proposition 7 Results

A detailed look at the Proposition 7 results is presented in Figure 5-11. The Output values can vary between -15 and 15 for this Proposition. As a result, the Firing Strength can vary between -75 and 75.

As shown in Table 5-1, this Proposition depends on both the Horizontal Separation and the Relative Ground Speed, with the results selected from the input which produces the lowest Membership. By comparing Figures 5-10 and 5-11, it can be seen that the Relative Ground Speed appears to produce the non-zero results in Figure 5-11.

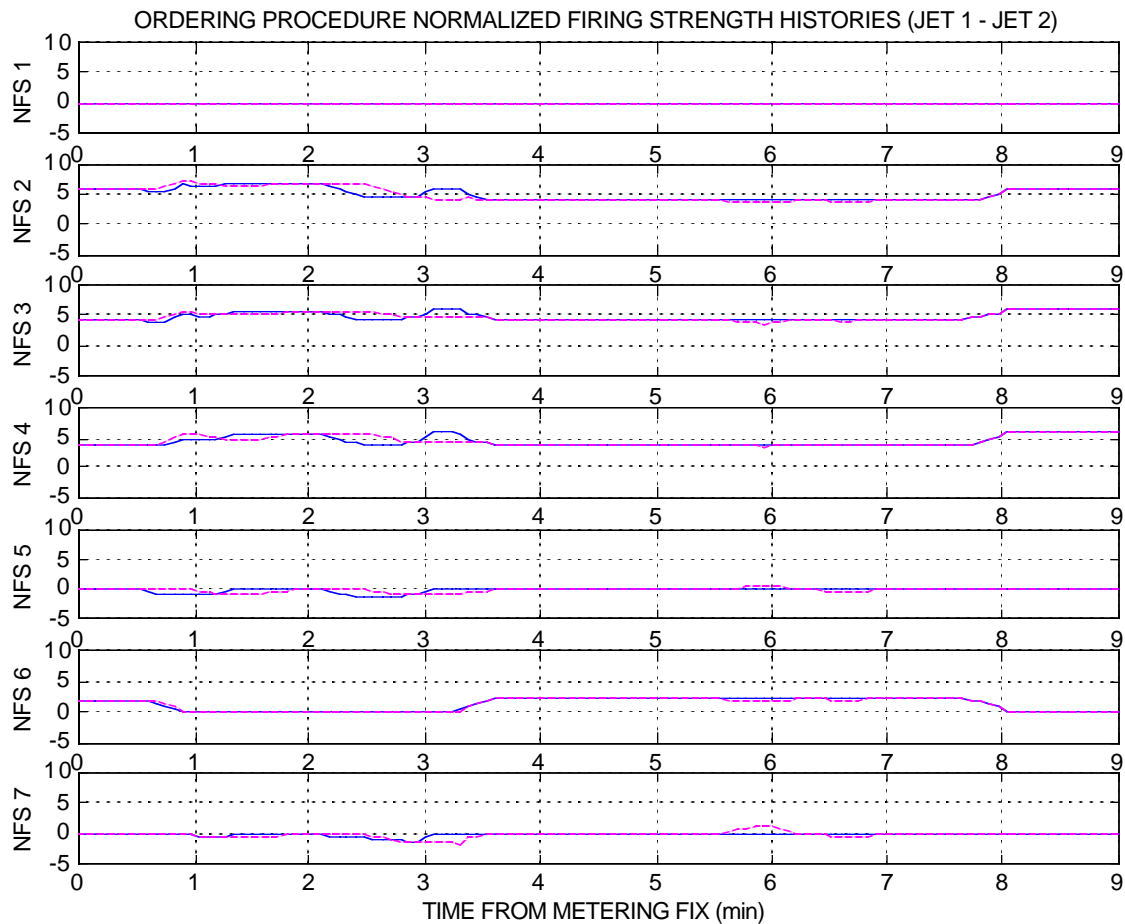


Figure 5-9. Ordering Procedure Normalized Firing Strength Histories
(Jet 1 ahead of Jet 2)

5.4.7 Ordering Procedure Results

Finally, when the individual Proposition results are combined to determine the Ordering Procedure results, the histories of Figure 5-12 are obtained. This figure shows that the total Output, total Weight, total Firing Strength, and the total Normalized Firing Strength are all positive. This confirms that the decision to have Jet 1 ahead of Jet 2 is the preferred order for these two aircraft.

If the Normalized Firing Strength history is examined, it varies approximately between 10 and 20. This Normalized Firing Strength history appears to lie in the neighborhood of Proposition Outputs of +7.5, +15, and +30. These Outputs correspond respectively to 'Marginally favored', 'Slightly favored' and 'Favored'. Hence, one can interpret the Procedure Normalized Firing Strength results to suggest that the order of Jet 1 ahead of Jet 2 is 'Slightly favored'.

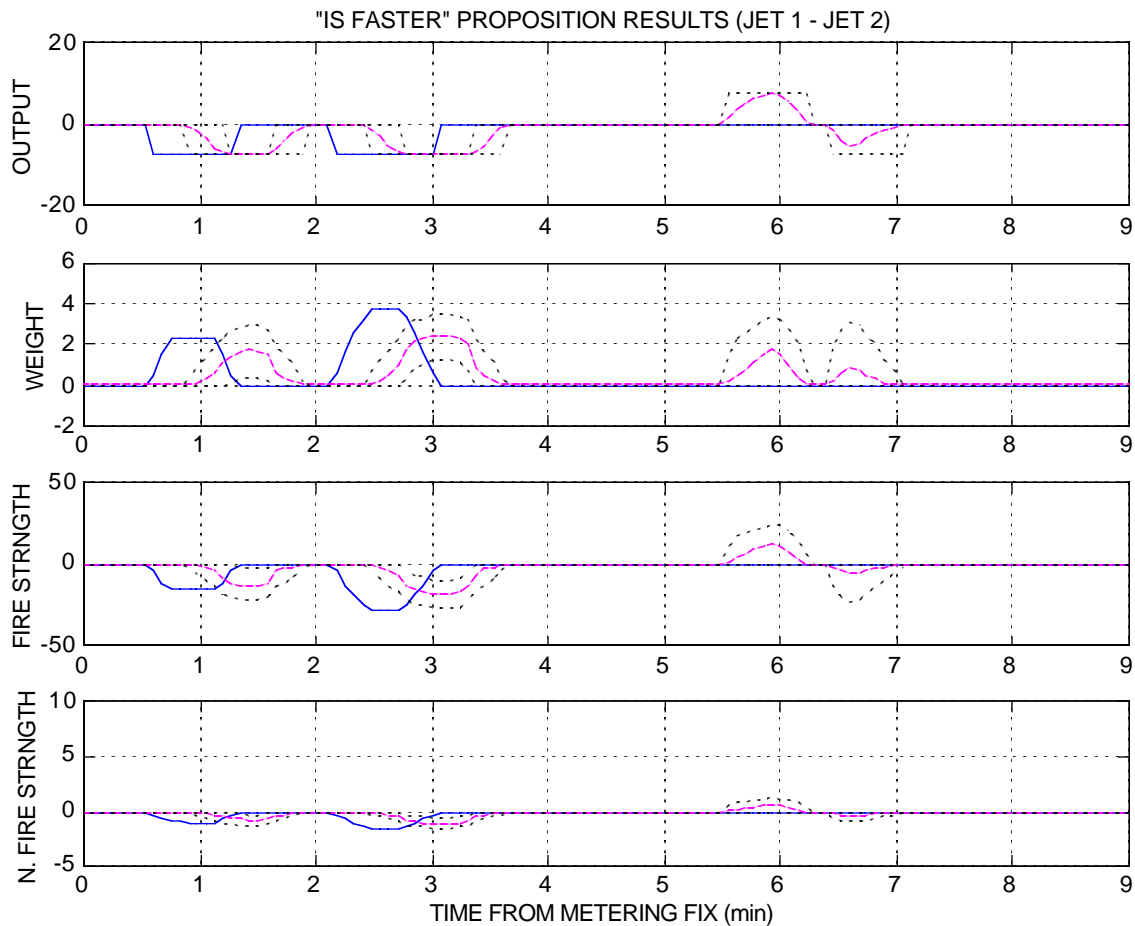


Figure 5-10. 'Is Faster' Proposition Results (Jet 1 ahead of Jet 2)

In summary, an Ordering Procedure scenario consisting of two in-track Jets has been evaluated. With the introduction of the ground speed tracking errors, the Ordering Procedure Firing Strength histories show some transient perturbations from the nominal. The net impact of the ground speed tracking errors on the ordering decision reached by this Procedure is not significant for this scenario.

Caution should be used in generalizing the results of this scenario whose primary purpose was to illustrate the features of the Ordering Procedure Performance Simulation. A larger number of additional scenarios should be investigated before more general conclusions can be reached about the performance of this Procedure. These scenarios would involve different flight path histories and different aircraft separations.

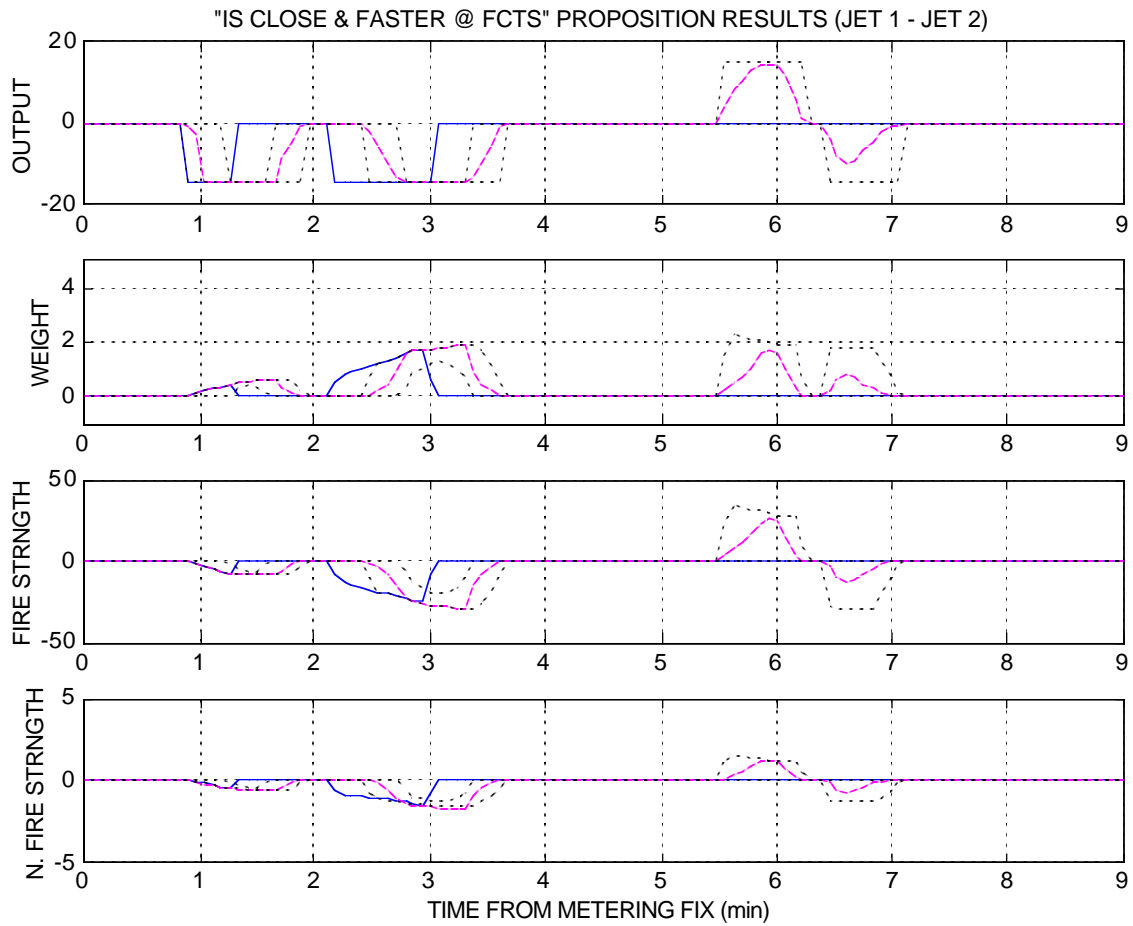


Figure 5-11. Ordering Procedure 'Is Close AND Faster' Proposition Results
(Jet 1 ahead of Jet 2)

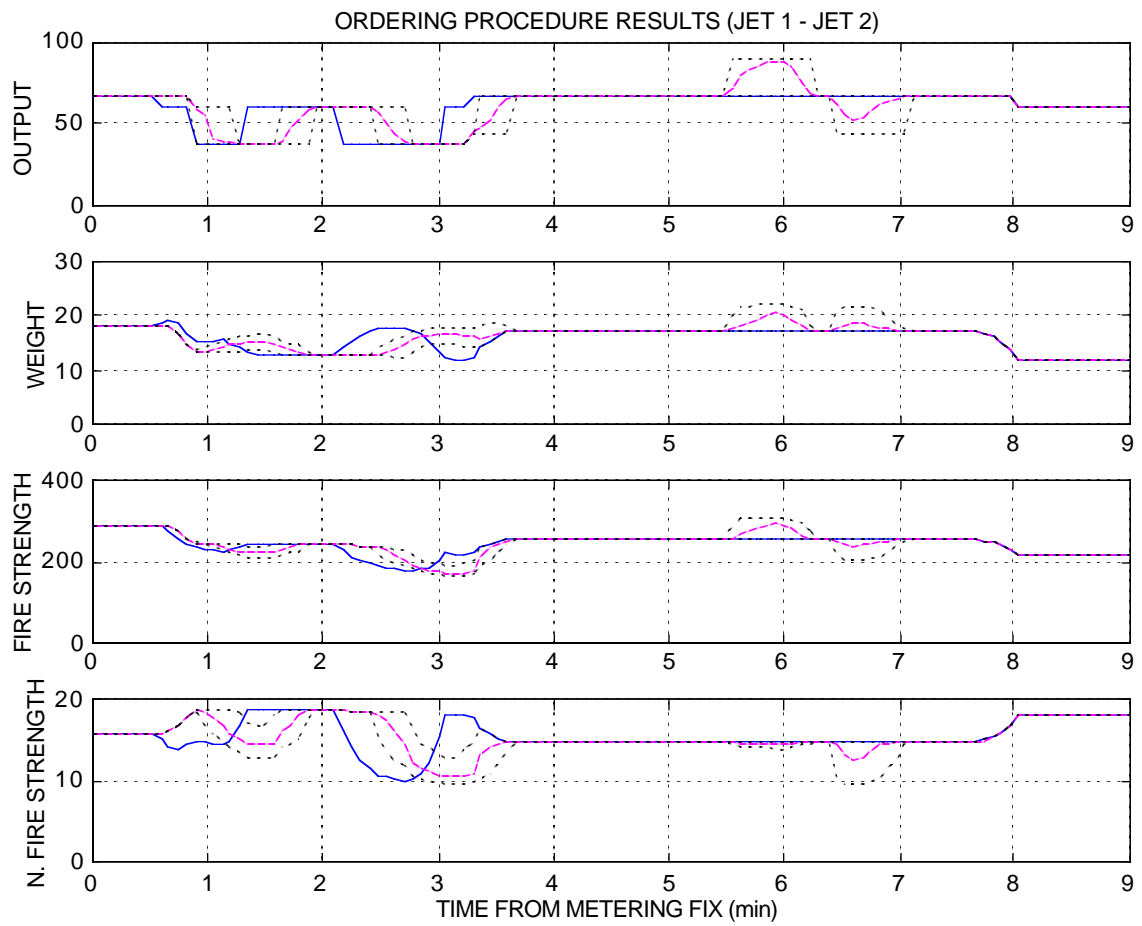


Figure 5-12. Ordering Procedure Results (Jet 1 ahead of Jet 2)

6.0 MERGING PROCEDURE PERFORMANCE SIMULATION RESULTS

In this Chapter, the performance of the Merging Procedure of a GENERAL-Type Spatial Constraint is evaluated. While the focus is on the Propositions which are directly or indirectly dependent on the ground speed errors, it is necessary to also model the remaining Propositions. For these latter Propositions, the inputs are assumed to be error-free relative to the ground speed dependent Propositions. By modeling all the Propositions for this Procedure, it is possible to determine the influence that the ground speed errors have on the Procedure decision. A listing of the performance simulation is presented in Appendix C.

6.1 Merging Procedure Performance Simulation

Similarly to the Ordering Procedure of a GENERAL-Type Spatial Constraint, the Merging Procedure consists of seven Proposition pairs as shown in Table 6-1. Now, however, in addition to the Relative Ground Speed input, the Normalized Separation Distance at FCTS Proposition input is also dependent on ground speed.

Table 6-1 Merging Procedure of a GENERAL-Type Spatial Constraint

| Number | Proposition | Input | Consequent | Output |
|--------|----------------------------------|---------------------------|-------------------------|--------|
| 1 | 'Significantly ahead at FCTS' | NSD_{FCTS} | 'Significantly favored' | +/-45 |
| 2 | 'Ahead at FCTS' | NSD_{FCTS} | 'Favored' | +/-30 |
| 3 | 'Slightly ahead at FCTS' | NSD_{FCTS} | 'Slightly favored' | +/-15 |
| 4 | 'Ahead at current position' | NSD_{TRK} | 'Slightly favored' | +/-15 |
| 5 | 'Faster at FCTS' | ΔV_G | 'Marginally favored' | +/-7.5 |
| 6 | 'Lower at current position' | Δh | 'Marginally favored' | +/-7.5 |
| 7 | ('Close') AND ('Faster at FCTS') | Δd & ΔV_G | 'Slightly favored' | +/-15 |

The FAST SL Performance Simulation which is used to evaluate the Merging Procedure of a GENERAL-Type Spatial Constraint is illustrated in Figure 6-1. This Performance simulation computes the individual Proposition expected Membership values, Outputs, Weights, and Firing Strengths and then combines these to determine the expected Procedure Output, Weight, and Firing Strength.

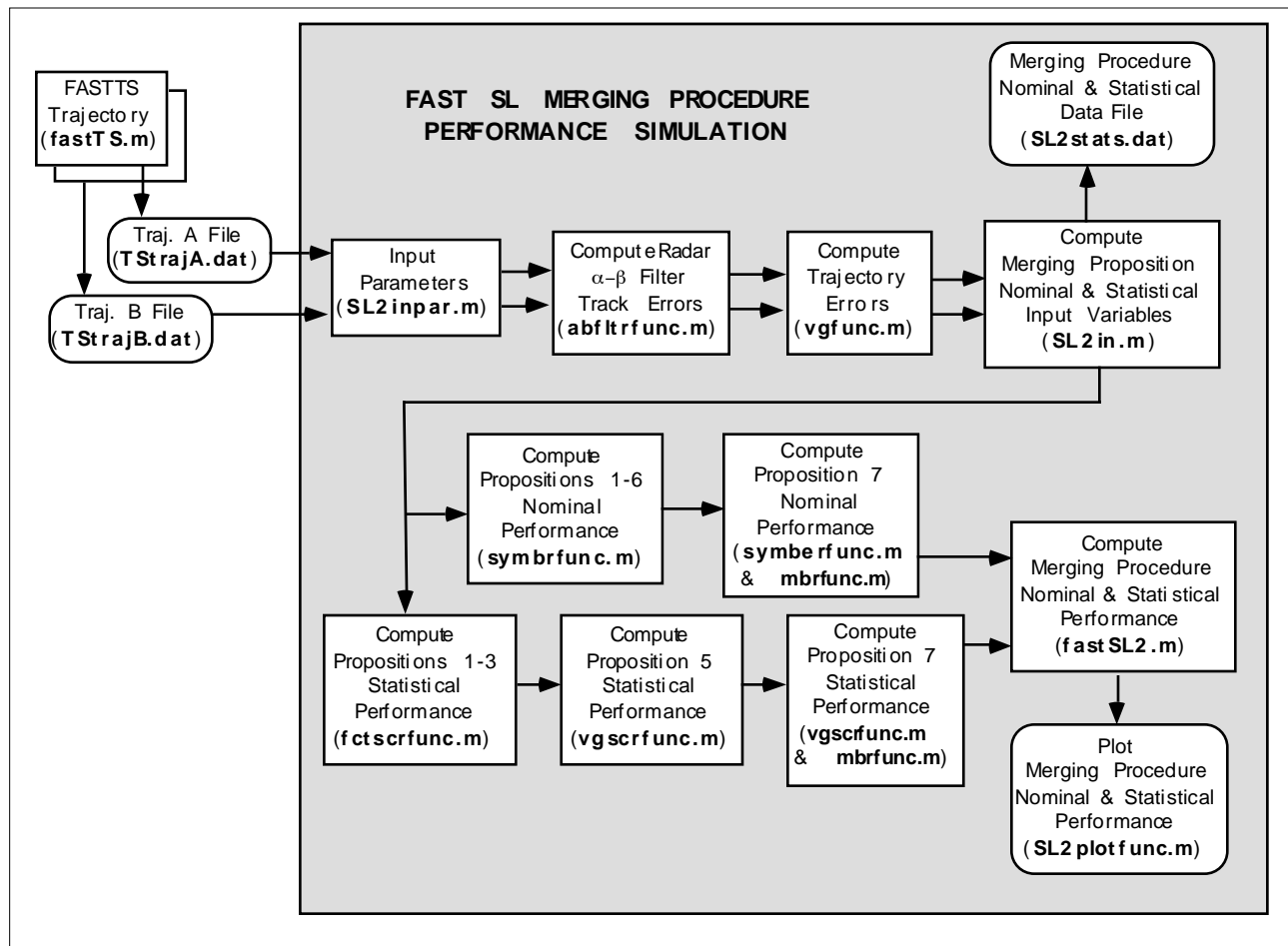


Figure 6-1. FAST SL Merging Procedure Performance Simulation

These Proposition calculations can be performed with both nominal (error-free) inputs or with statistical inputs. In the latter case, only the inputs which depend directly or indirectly on the Relative Ground Speed errors are substituted in place of their nominal values. It is assumed that the remaining inputs are relatively close to nominal when compared to those ground speed dependent inputs.

By comparing the performance of this Procedure under nominal as well as perturbed conditions, the degradation of the Procedure due to the Relative Ground Speed errors can be determined. Under this study, the principal figures of merit which are used are the expected (mean) Proposition Membership, Output, Weight, and Firing Strength. The effects of the corresponding dispersions (standard deviations) are only used to loosely determine a confidence interval about the estimate. Hence, a 95% confidence interval corresponds to ± 2 sigma, assuming that the statistics are approximately Gaussian.

Examining Figure 6-1 more closely, the simulation requires the nominal trajectory histories for the two aircraft whose merging sequence is determined. These trajectories can be obtained using the fastTS.m MATLAB simulation (Appendix A) or they can be

obtained from some other source. In the latter case, for instance, actual aircraft trajectories can be used, so long as the same trajectory variables and ASCII format is used for the data files.

These trajectories as well as all the key control parameters are entered into the SL2inpar.m MATLAB script. Using these parameters and trajectory histories, the radar tracking error histories are computed in abfltrfunc.m. With the radar tracking error histories, the aircraft trajectory error histories are computed in vgfunc.m. These trajectory error histories are then combined in SL2in.m to obtain the nominal and statistical relative input variable histories for the seven Proposition pairs.

Using these nominal relative variable histories, Propositions 1 - 6 are evaluated using symbrfunc.m while Proposition 7 uses symbrfunc.m and mbrfunc.m. For the Merging Procedure, there are now two sets of ground speed dependent input variables. These are the Relative Ground Speed and the NSD_{FCTS} for the two aircraft. Using the NSD_{FCTS} statistics, Propositions 1 - 3 are evaluated in fctscrfunc.m. The Relative Ground Speed statistics are evaluated for Proposition 5 using vgscrfunc.m and in Proposition 7 using mbrfunc.m and vgscrfunc.m.

The individual Proposition results are combined in fastSL2.m to determine the Merging Procedure decision based on the Procedure composite Firing Strength. By using the nominal Proposition results, the nominal Procedure decision results are obtained. Substituting the statistical Proposition results for the corresponding nominal Proposition results, leads to the statistical Procedure decision. These results are plotted using SL2plotfunc.m.

6.2 Nominal Trajectories

The nominal trajectories, which are used, correspond to the same scenario presented in Section 3.2. This corresponds to the approach to Dallas-Ft. Worth Runway 18R from the southwest metering fix. In this section the focus are on the nominal jet and turboprop trajectories, since they present a merging scenario for this Procedure. Specifically the nominal jet and turboprop trajectories are time shifted (biased) to produce the desired separation for the merging scenario onto to the Downwind flight path segment. This corresponds to Jet 2, of the two jet scenario used in Chapter 5, positioned behind the Turboprop at their nominal FCTS, as illustrated in Figure 6-2. Under this scenario, the Turboprop is ahead of the Jet by 3 nm. at FCTS. This corresponds to the nominal minimum required separation distance for these two aircraft and their flight sequence.

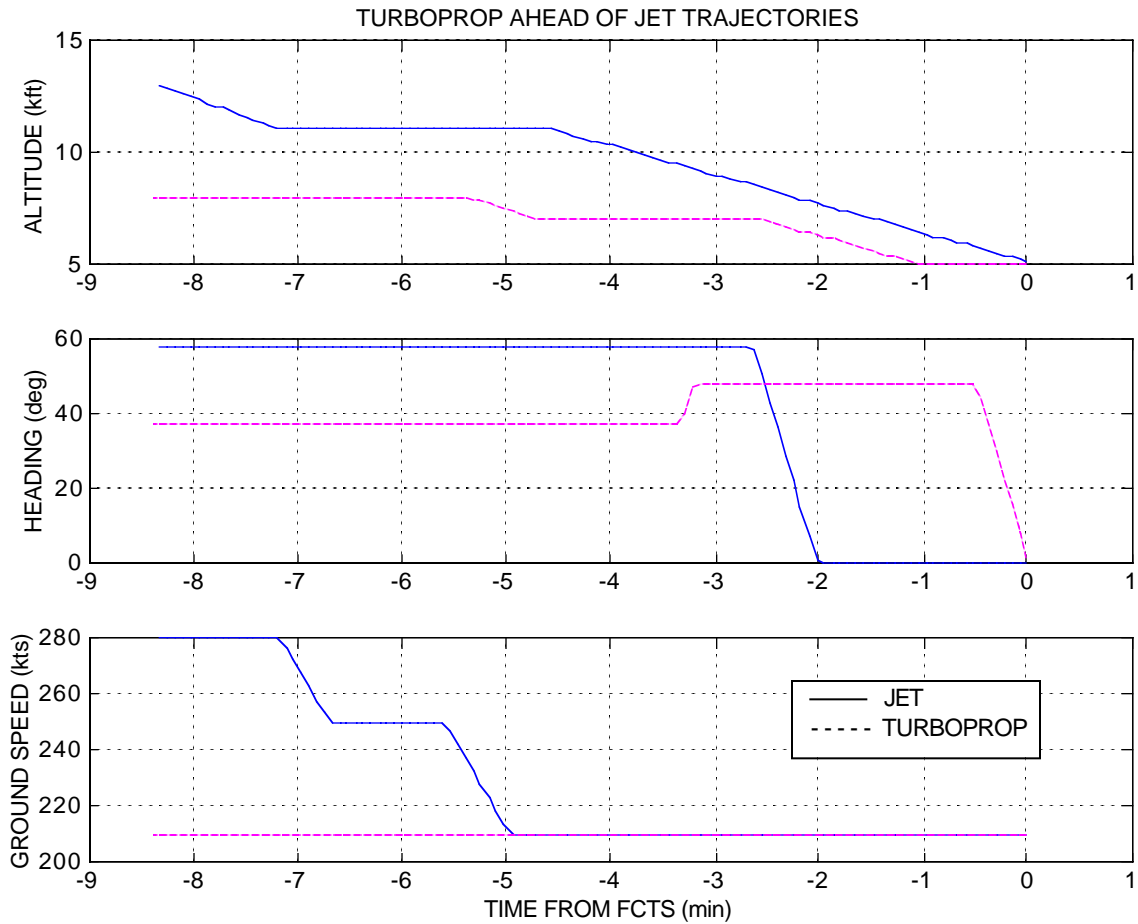


Figure 6-2. FAST TS Time Histories (Turboprop ahead of Jet)

6.3 Merging Procedure Input Statistics

In this section, the FAST SL Proposition input error statistics are presented. First the nominal relative trajectory variables are computed. Then the estimation errors are combined with the nominal relative trajectory variables to obtain the estimated relative trajectory variables. Since the only ground speed dependent variables are the NSD_{FCTS} and Relative Ground Speed, the focus is on their error statistics.

6.3.1 NSD_{FCTS} Statistics

The NSD_{FCTS} nominal and error statistics are illustrated in Figure 6-3 for the Turboprop ahead of the Jet. The time interval is the time from the metering fix to their FCTS. The estimate is also shown with the 95% confidence interval.

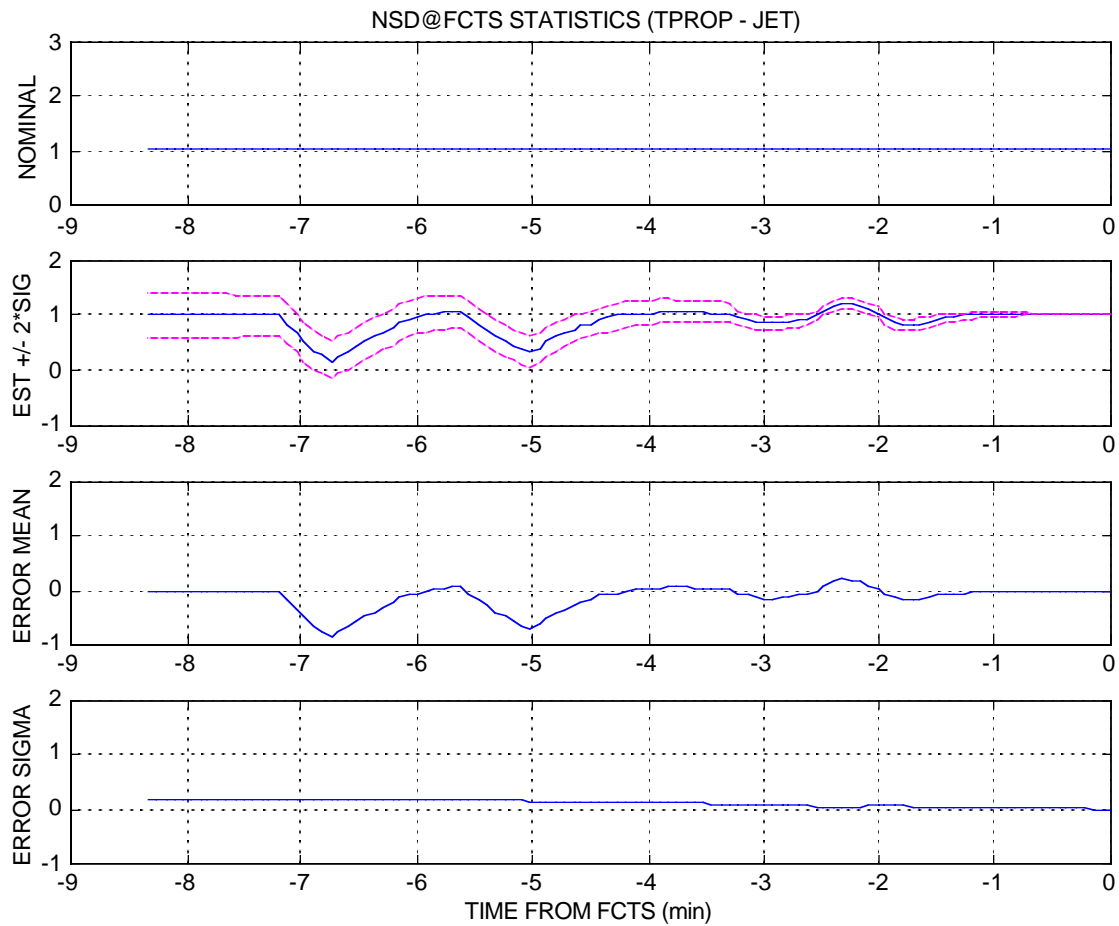


Figure 6-3. NSD_{FCTS} Statistics (Turboprop ahead of Jet)

An examination of this figure shows that the nominal NSD_{FCTS} has a constant value of 1 while the estimate varies approximately between 0 and 1. The individual spikes observed in the estimate and the mean statistics are produced by one or the other aircraft undergoing a speed reduction or a heading change.

6.3.2 Relative Ground Speed Statistics

In this section the focus is on the Relative Ground Speed estimation error at the FCTS for the Turboprop ahead of the Jet trajectories. This case is illustrated in Figure 6-4.

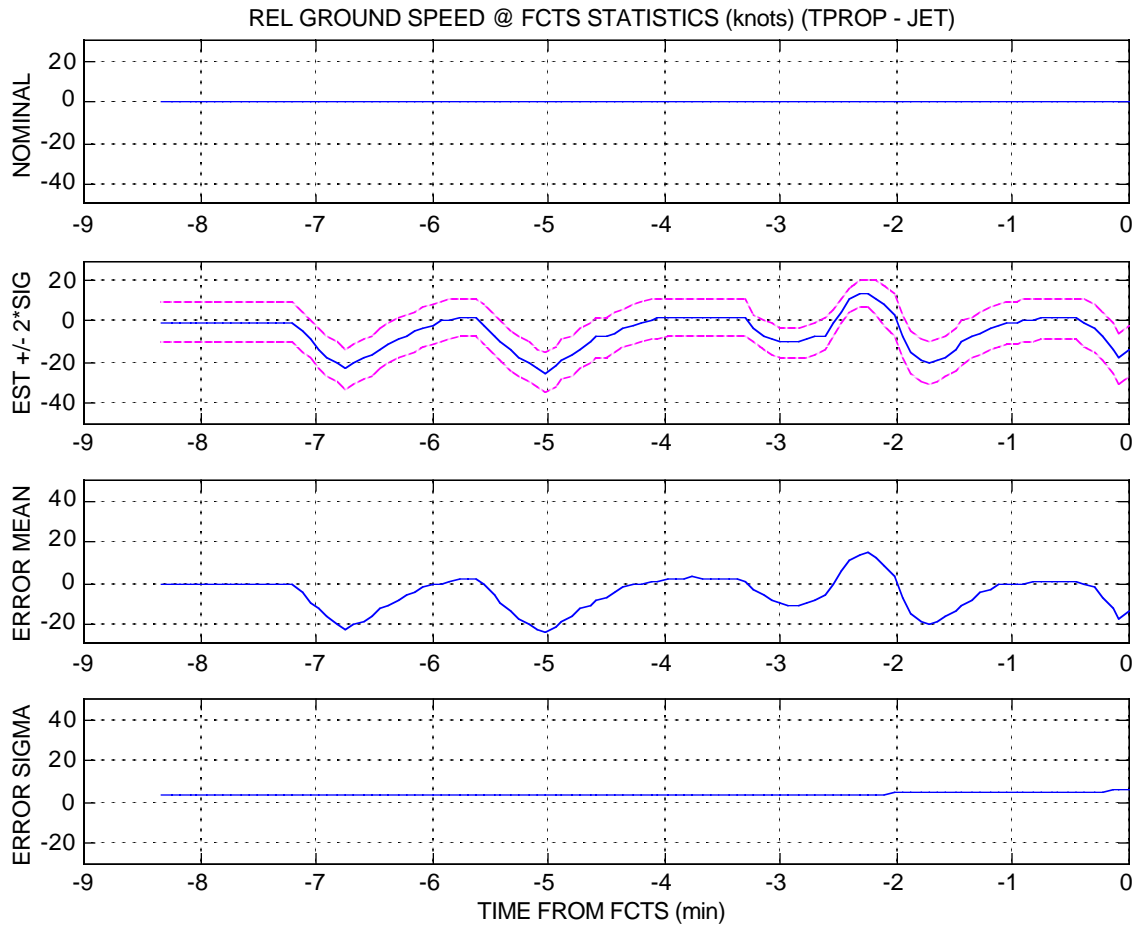


Figure 6-4. Relative Ground Speed Statistics
(Turboprop ahead of Jet)

Examining this figure, it can be seen that the nominal Relative Ground Speed at FCTS is zero. With the addition of the mean Relative Ground Speed error to the nominal, the Relative Ground Speed estimate is obtained. This estimate, which also includes the 95% confidence interval for this estimate, is seen to vary roughly between ± 20 knots. The 95% confidence interval about this estimate, extends the range to roughly -30 to +20 knots. The individual spikes observed in the estimate and mean error histories correspond to either one or the other aircraft undergoing a speed reduction maneuver or a heading change.

As discussed in Section 2.3.3, the Relative Ground Speed at FCTS estimate, is nominally the current Relative Ground Speed estimate. However, if there is a nominal speed reduction maneuver between the current time and t_{FCTS} , the current Relative Ground Speed at FCTS estimate is the nominal Relative Ground Speed at FCTS. This follows from the assumption that the current estimate of the ground speed will be flown by each aircraft until a nominal speed reduction is made, since FAST TS uses only nominal maneuver profiles. This reduction of the Relative Ground Speed at FCTS errors by

nominal future speed maneuvers has not been considered in these figures, making these results conservative. However, these figures also have not considered other ground speed errors such as unpredicted winds and flight technical (pilot steering) errors.

6.4 Merging Procedure Decision Statistics

In this section, the decision statistics for the Merging Procedure of a GENERAL-Type Spatial Constraint are presented. While the principal statistic are the Proposition and the Procedure expected Firing Strength, the expected Membership, Output, and Weight are also presented.

6.4.1 Membership Histories

The input histories to the Propositions of this Merging Procedure are presented in Figure 6-5 for the Turboprop ahead of the Jet. These indicate very similar results for the nominal cases, shown with the solid lines, and more variability for the statistical estimates, shown with dashed lines. Note that the NSD_{FCTS} and NSD histories in Figure 6-5 are based on a normalization factor of 3 nm.

The expected Membership histories for the Turboprop ahead of the Jet are shown in Figure 6-6. In this figure, the nominal Membership histories are shown by the solid curves while the estimates are shown with the dashed curves. While most of the estimate histories are fairly close to their corresponding nominal histories, the third Proposition does show some significant deviations from the nominal.

6.4.2 Output Histories

The expected Output histories are shown in Figure 6-7. Note that while the nominal Output can only assume three discrete values (zero, +nominal, or -nominal), the expected Output values can vary continuously between these extremes. This arises from the use of the probability weighting.

6.4.3 Weight Histories

Figure 6-8 presents the nominal and estimated Weight histories for these seven Propositions. The general shapes of these histories are seen to be similar to the Membership histories of Figure 6-6.

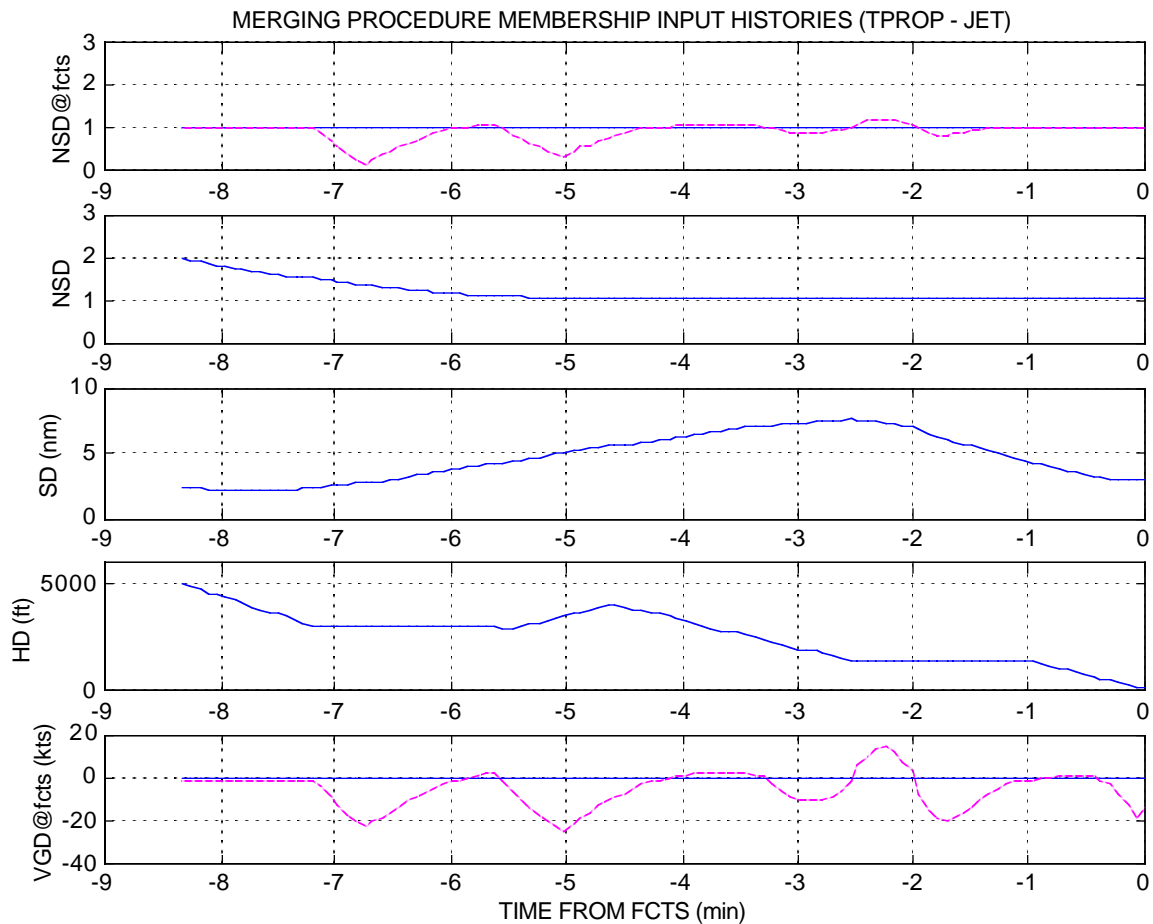


Figure 6-5. Merging Procedure Input Histories
(Turboprop ahead of Jet)

6.4.4 Firing Strength Histories

The expected Firing Strength histories for this scenario are presented in Figure 6-9. The nominal Firing Strength histories can be obtained as the product of the nominal Output and the nominal Weight histories. However, the estimated Firing Strength history cannot, in general, be obtained as the product of the estimated Output history and the estimated Weight history, as is shown in Chapter 4. If the standard deviations of the Output and the Weight are small, then their estimated Firing Strength is approximately equal to the product of the estimated Output and the estimated Weight. Due to these direct or approximate relationships, the Firing Strength histories tend to have similar shapes as the Weight histories.

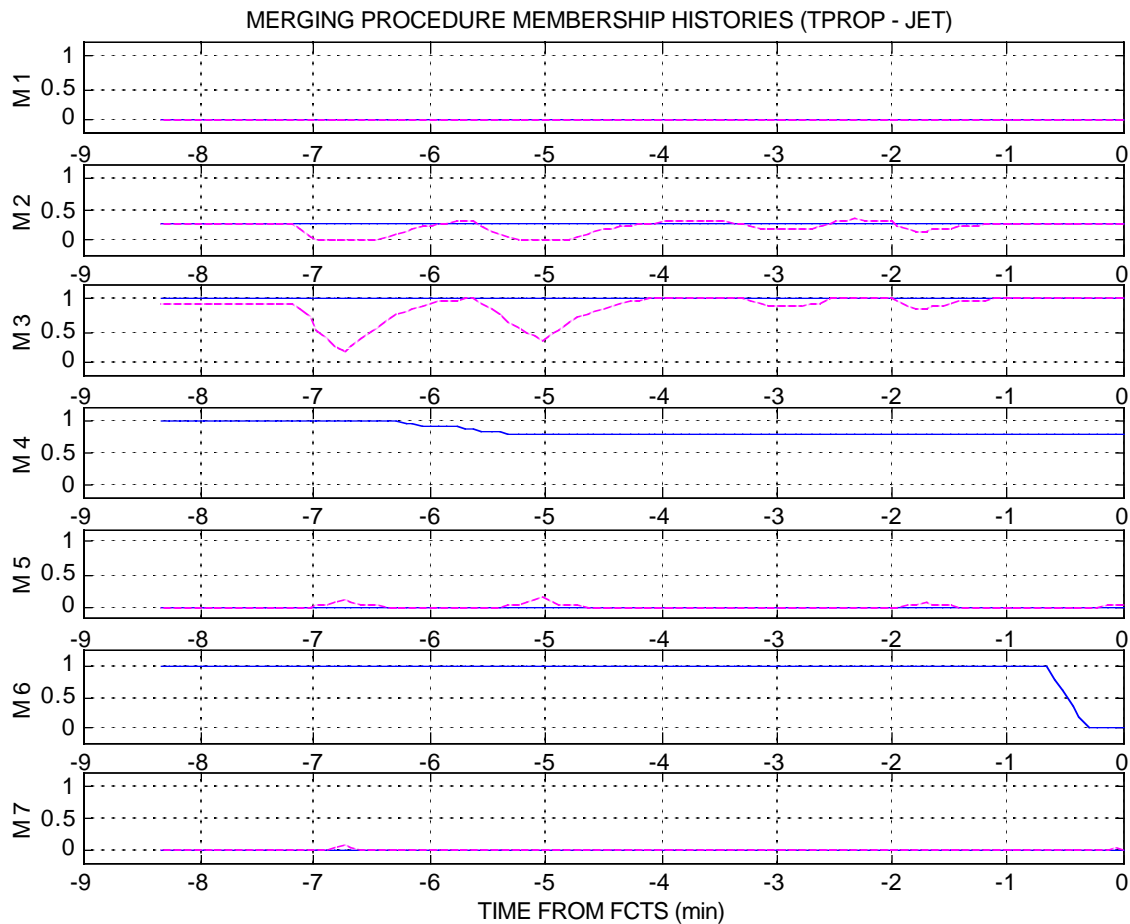


Figure 6-6. Merging Procedure Membership Histories
(Turboprop ahead of Jet)

In Figure 6-9 are presented the nominal and statistical Normalized Firing Strength histories. These are obtained by normalizing the nominal Firing Strength histories by the sum of the seven Proposition nominal Weight histories. In similar fashion, the statistical Normalized Firing Strength histories are obtained by normalizing the statistical Firing Strength histories by the sum of the seven Proposition statistical or nominal Weights.

The Normalized Firing Strength can be viewed as a Weighted Output. As a result, it is possible to examine its value and assign a similar significance as the Output values have. Hence values near +7.5 can be thought of as 'Marginally favoring the current aircraft sequence' while values near +45 can be thought of as 'Significantly favoring the current aircraft sequence.' Similar arguments apply for negative values in which case the reverse aircraft sequence is preferred. In addition, the nominal Normalized Firing Strength is not limited to discrete values, unlike the nominal Output. An examination of Figure 6-9 shows that most of the Normalized Firing Strength histories lie between zero and 10.

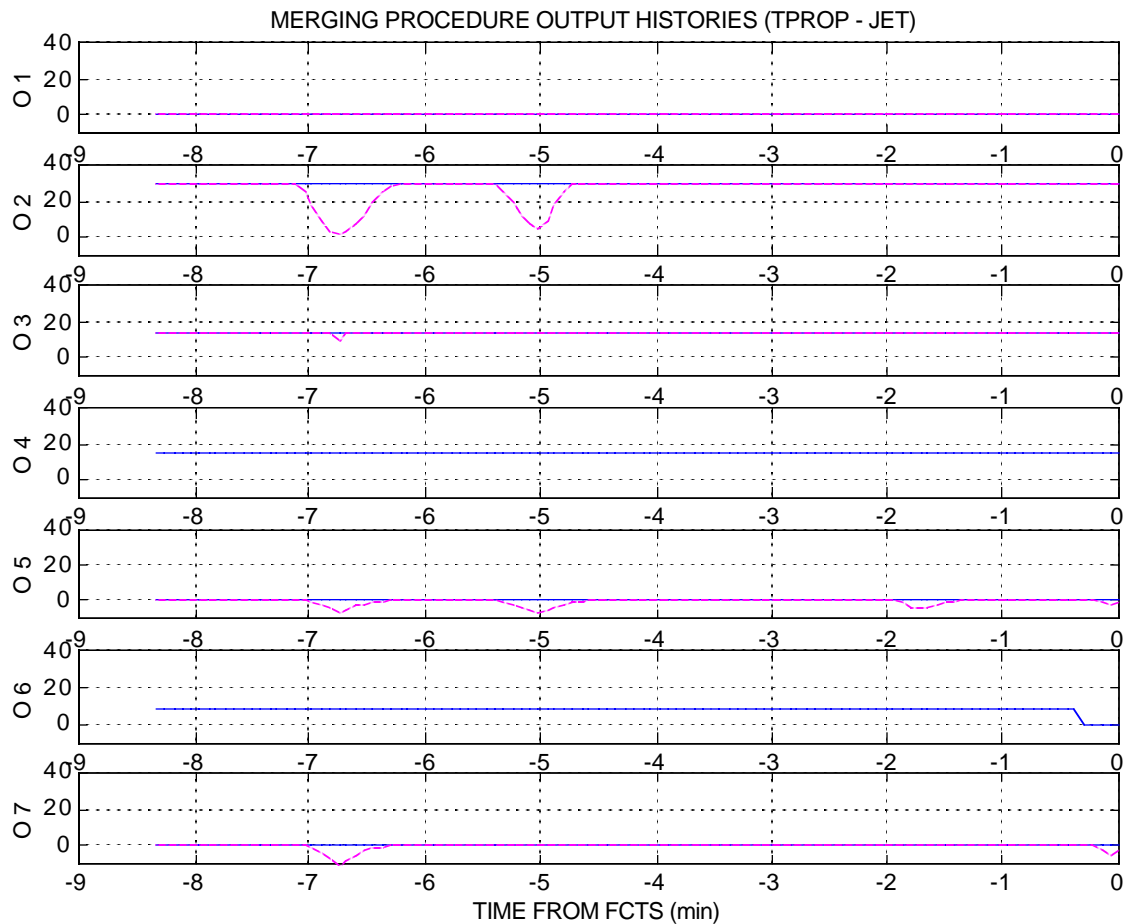


Figure 6-7. Merging Procedure Output Histories
(Turboprop ahead of Jet)

6.4.5 Proposition 1 - 3 Results

In the next five figures, a close look at the Proposition results is presented for those Propositions which rely on ground speed dependent inputs. These are Propositions 1 - 3, 5, and 7. In these figures, the nominal histories are shown with solid curves while the histories of the estimates are shown with dashed curves. In addition, the 95% confidence interval curves are shown as dotted curves.

In Figures 6-11 through 6-13 are presented the results for Propositions 1 through 3. Examining these NSD_{FCTS} Proposition figures, it can be seen that the results for Proposition 1 are zero while the Proposition 2 and 3 results are zero or positive. For Proposition 3, however, the 95% confidence limits show some brief periods of negative Output, Firing Strength, and Normalized Firing Strength results.

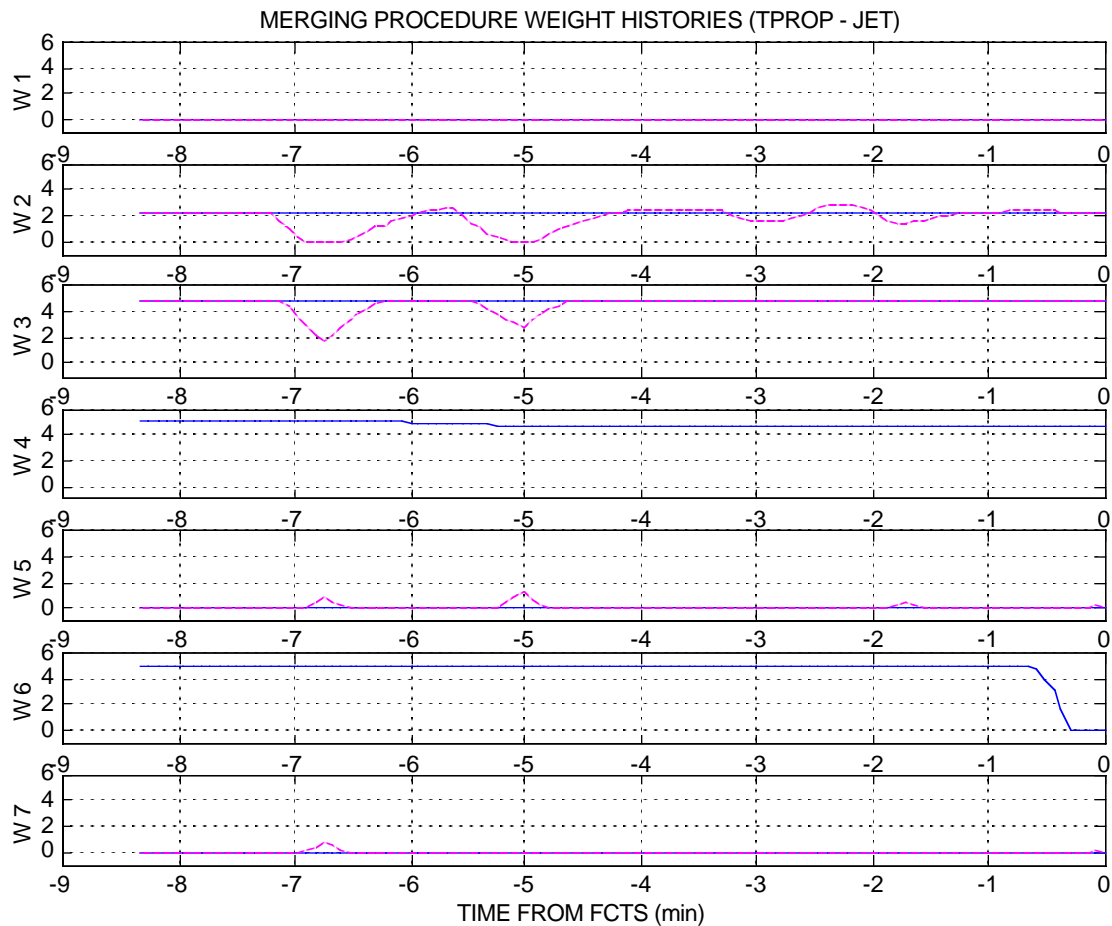


Figure 6-8. Merging Procedure Weight Histories
(Turboprop ahead of Jet)

6.4.6 Proposition 5 and 7 Results

In Figures 6-14 and 6-15 are presented the results for the Propositions 5 and 7 which both depend on the Relative Ground Speed. In general, the statistical results do not appear to differ significantly from the corresponding nominal results.

6.4.7 Merging Procedure Results

In Figure 6-16 are presented the Merging Procedure results. Examining the Normalized Firing Strength histories for this Procedure, it can be seen that the nominal and statistical results vary between 10 and 20. In the context of the Proposition Output of +15, these results suggest that the Turboprop 'Is slightly favored' to merge ahead of the Jet onto the Downwind flight path segment.

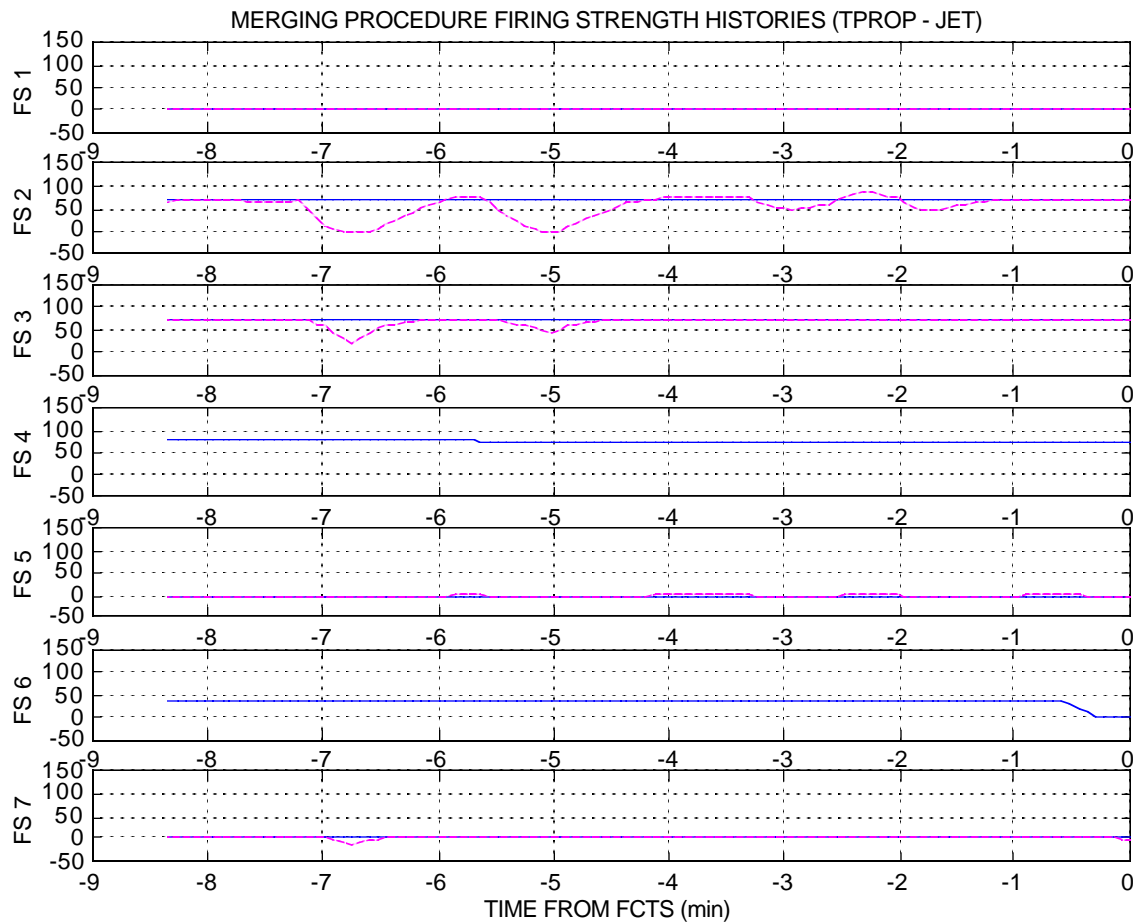


Figure 6-9. Merging Procedure Firing Strength Histories
(Turboprop ahead of Jet)

This analysis only included ground speed tracking errors. It did not consider additional FAST TS modeling errors due to the use of simplified dynamics models. Also the effects of wind prediction errors and pilot steering errors were not considered.

The perturbations, however, were insufficient to lead to an incorrect Merging Procedure decision. An incorrect Merging Procedure decision would have resulted if the expected Normalized Firing Strength estimates had switched polarity and had reached a value whose magnitude was at least 7.5. For this specific scenario, an incorrect Procedure decision would have been reached if the Normalized Firing Strengths reached values less than -7.5. This limit has been selected in the design of the FAST SL to add some hysteresis and avoid limit cycling back and forth between small positive and negative Normalized Firing Strength results.

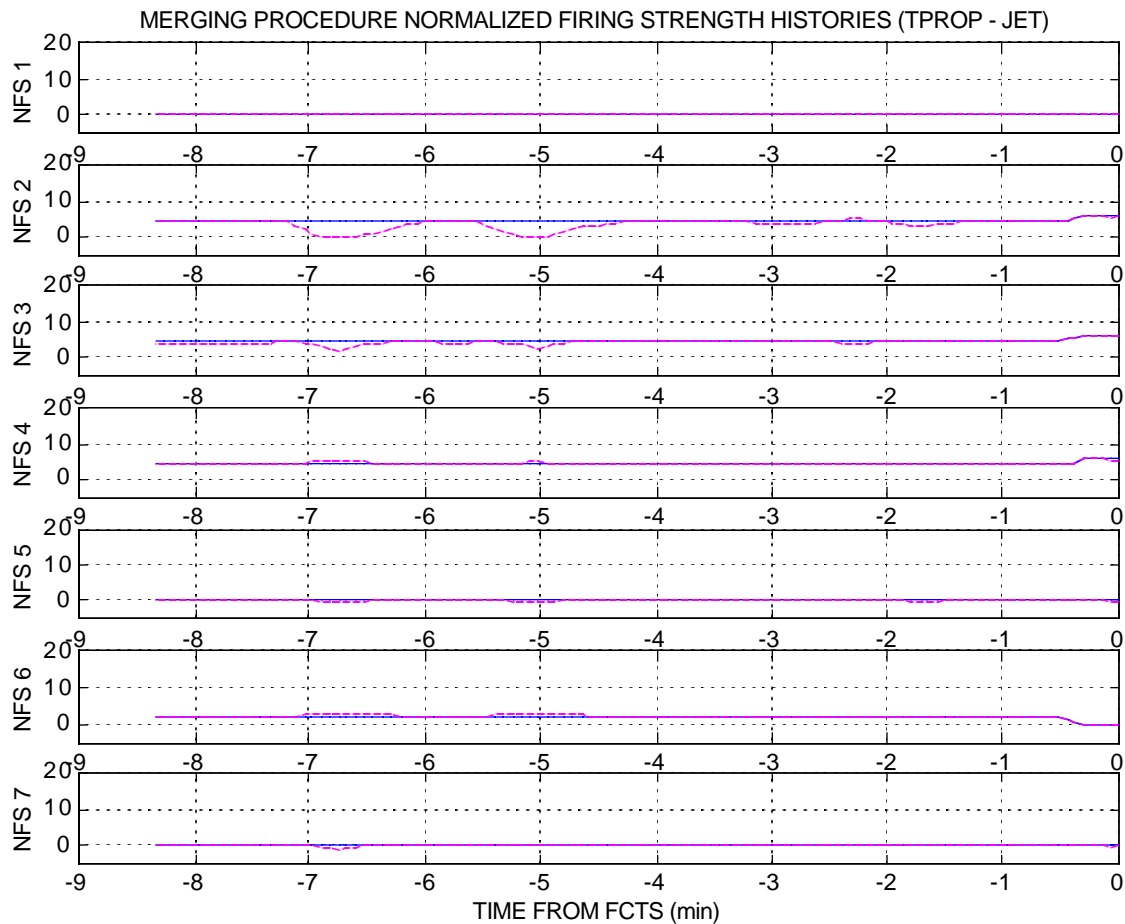


Figure 6-10. Merging Procedure Normalized Firing Strength Histories (Turboprop ahead of Jet)

If the nominal separation distances between the two sets of aircraft were increased, the nominal Normalized Firing Strength is expected to increase. Hence, it would be even less likely that the expected Normalized Firing Strength will switch polarity. However, if the nominal separation distances are less than nominal between the two sets of aircraft, the nominal Normalized Firing Strength is expected to decrease. Under these conditions it may be more likely that the Relative Ground Speed errors may lead to a polarity change in the expected Normalized Separation Distance which is sufficient to lead to an incorrect Merging Procedure decision. This would have to be investigated further.

In summary, a Merging Procedure scenario consisting of a Turboprop followed by a Jet aircraft has been evaluated. With the introduction of the ground speed tracking errors, the Merging Procedure Firing Strength histories show some transient perturbations from the nominal. The net impact of the ground speed tracking errors on the merging decision reached by this Procedure is not significant for this scenario.

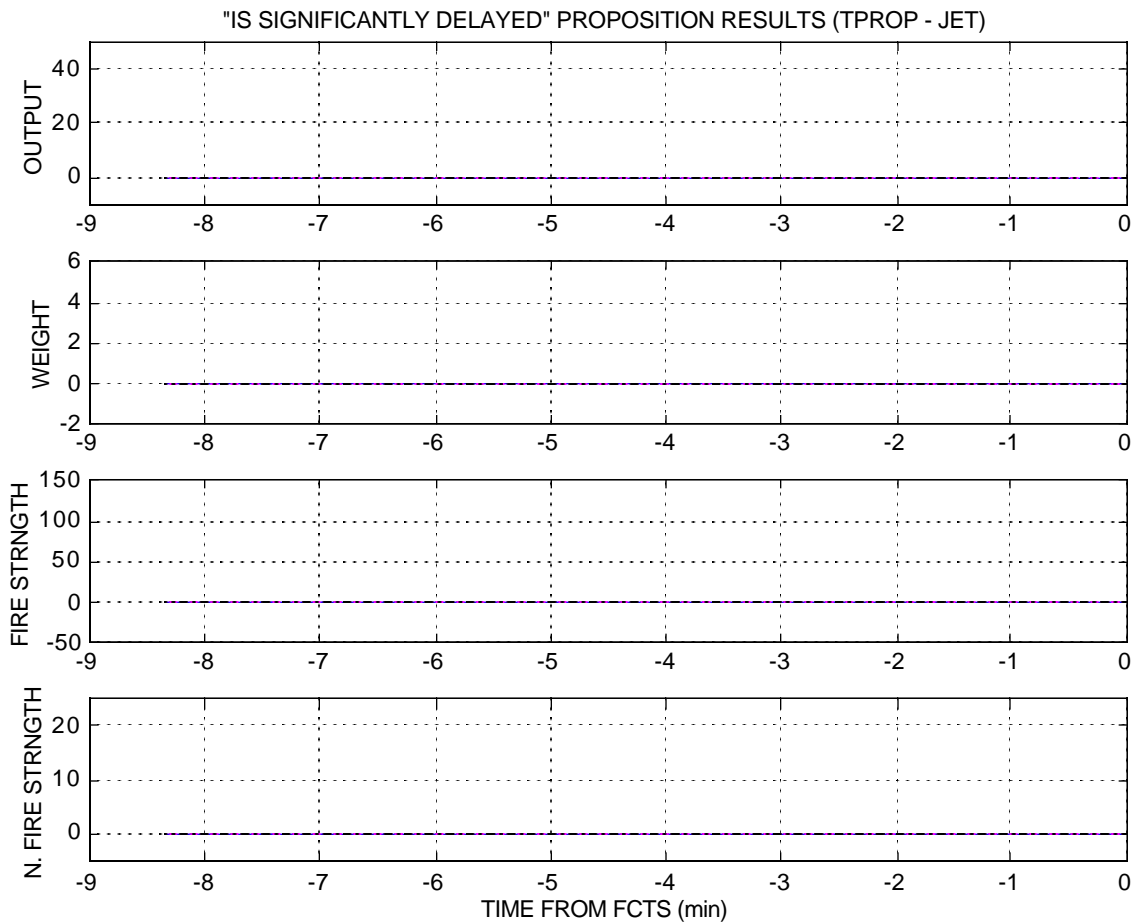
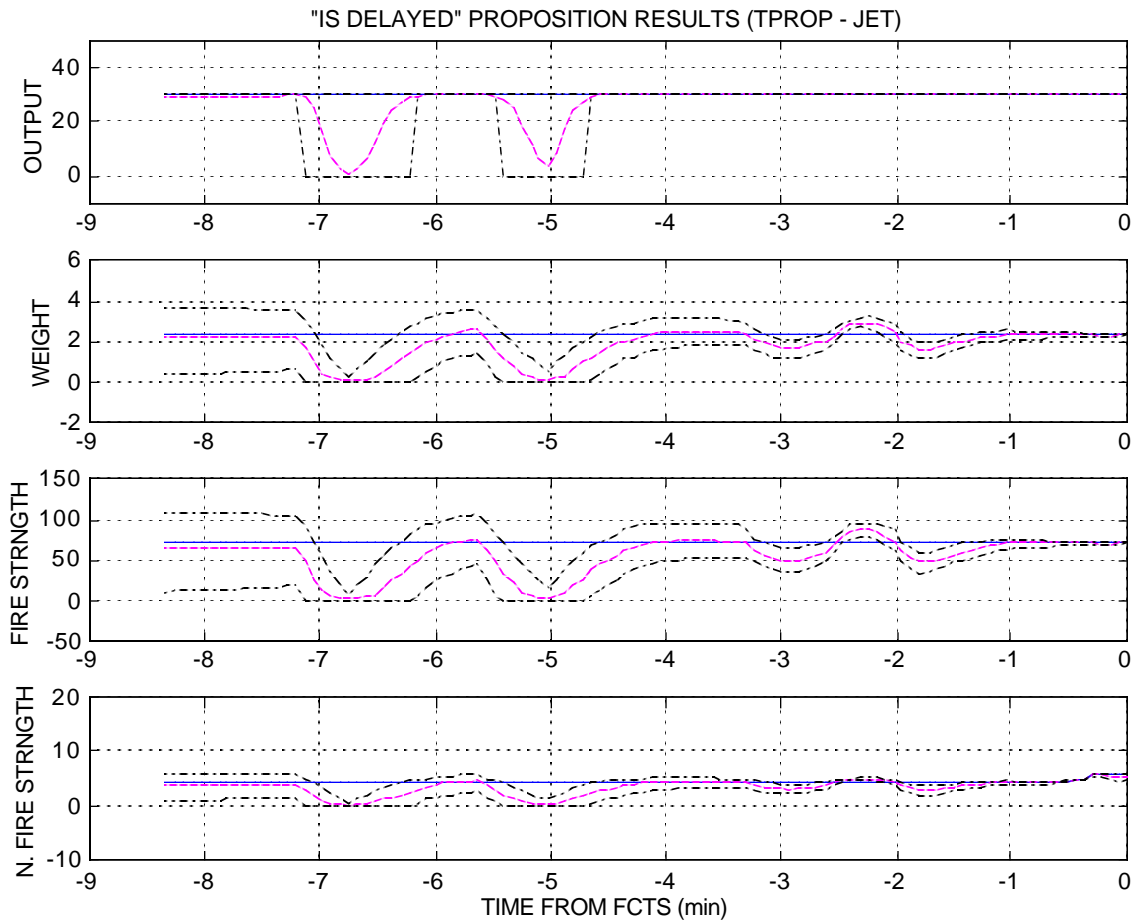


Figure 6-11. 'Is Significantly Delayed' Proposition Results
(Turboprop ahead of Jet)

Caution should be used in generalizing the results of this scenario whose primary purpose was to illustrate the features of the Merging Procedure Performance Simulation. A larger number of additional scenarios should be investigated before more general conclusions can be reached about the performance of this Procedure. These scenarios would involve different flight path histories and different aircraft separations.



**Figure 6-12. 'Is Delayed' Proposition Results
(Turboprop ahead of Jet)**

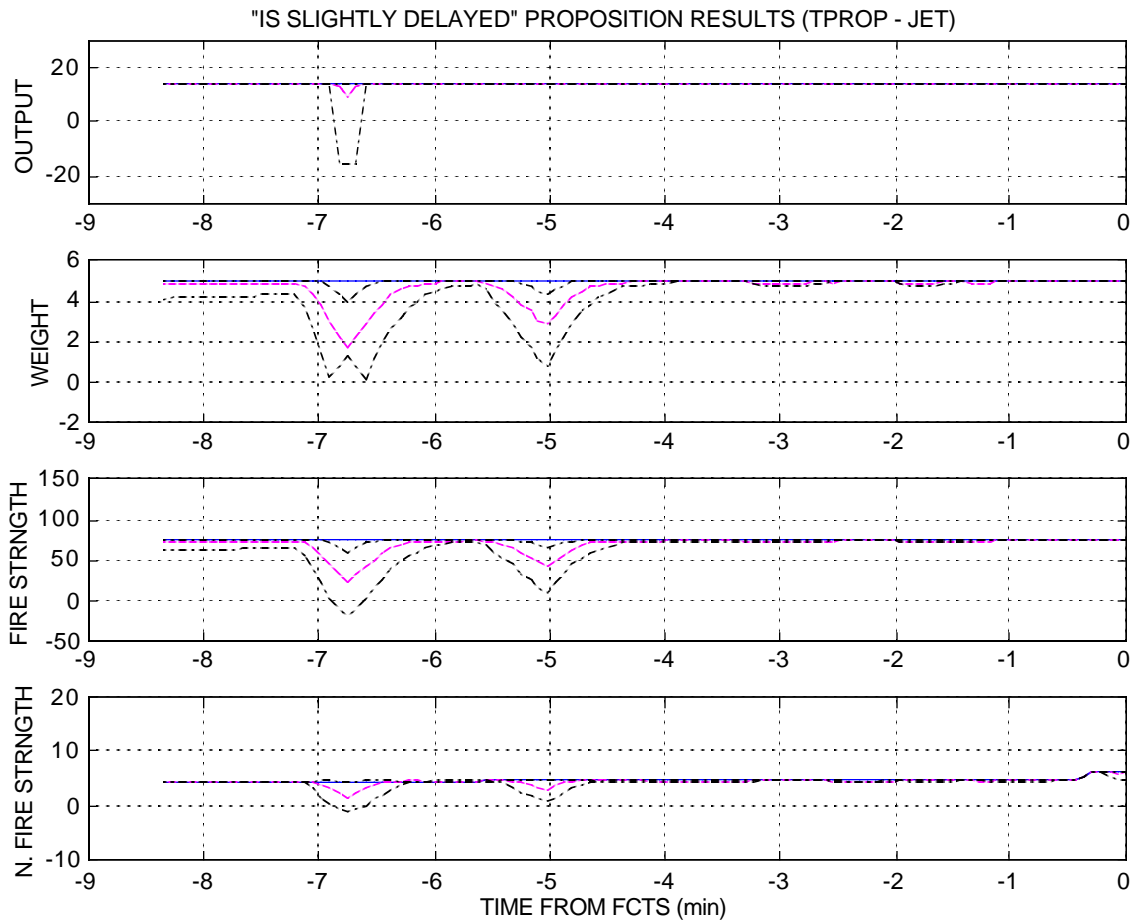


Figure 6-13. 'Is Slightly Delayed' Proposition Results
(Turboprop ahead of Jet)

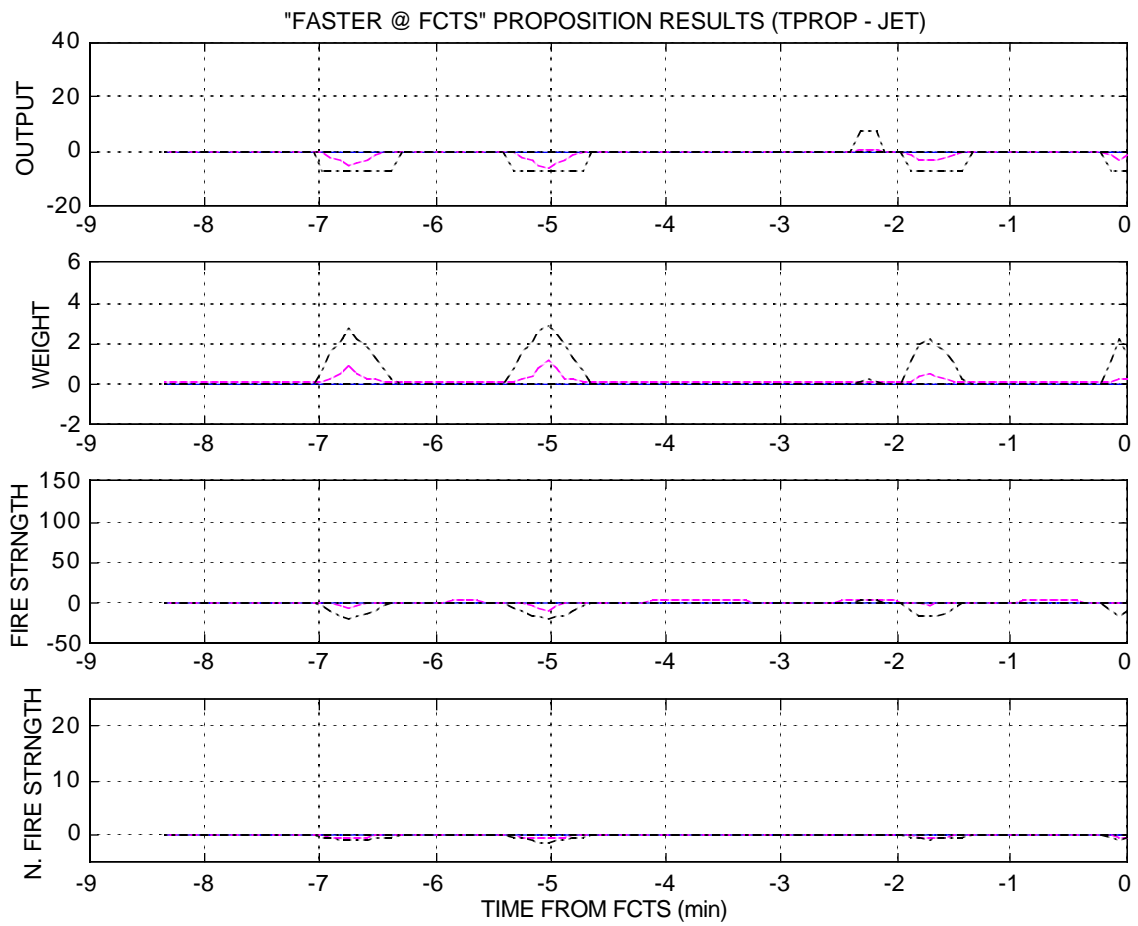


Figure 6-14. 'Is Faster' Proposition Results
(Turboprop ahead of Jet)

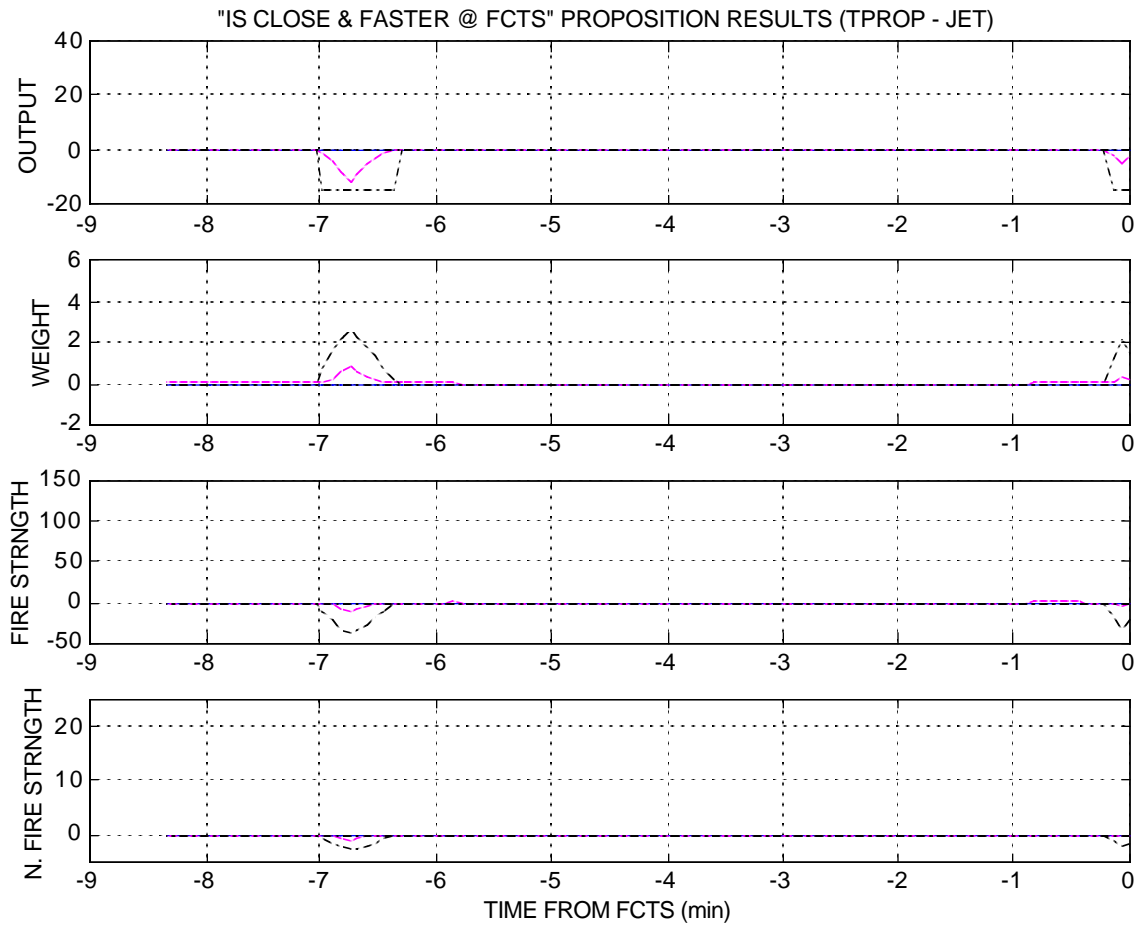


Figure 6-15. 'Is Close' AND 'Is Faster at FCTS' Proposition Results
(Turboprop ahead of Jet)

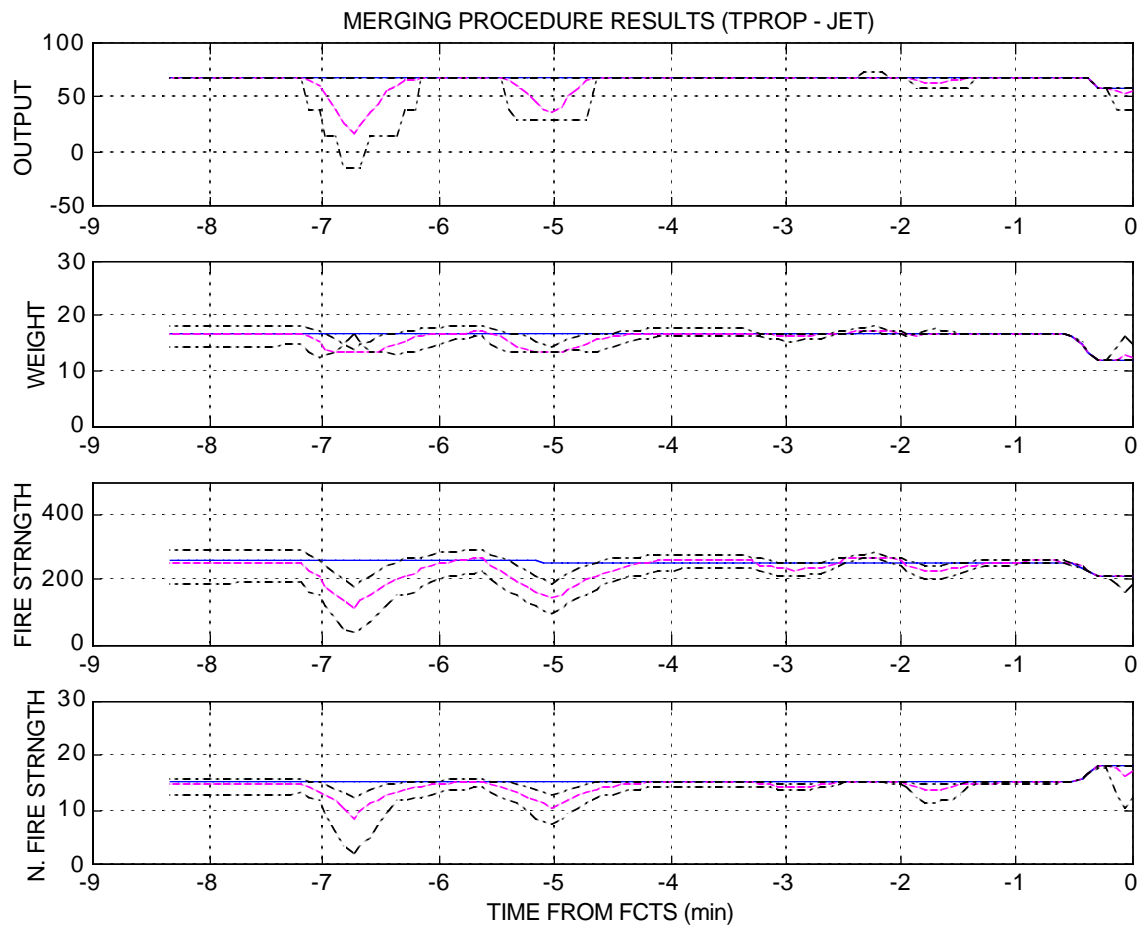


Figure 6-16. Merging Procedure Results
(Turboprop ahead of Jet)

7.0 REFERENCES

1. Anon, "Center-Tracon Automation System Description to Support an Operational Concept Document," Seagull Technology, Inc., 90112-03, November 1990
2. Bergh, C.P., Krzeczowski, K.J., and Davis, T.J., "TRACON Aircraft Arrival Planning and Optimization through Spatial Constraint Satisfaction," Air Traffic Control Quarterly, Vol. 3(2), 1995.
3. Davis, T.J., Krzeczowski, K.J., & Bergh, C., "The Final Approach Spacing Tool," 13th IFAC Symposium on Automatic Control in Aerospace, Palo Alto, CA, September 1994.
4. Davis, T.J., et al, "Operational Test Results of the Passive Final Approach Spacing Tool," 8th IFAC Symposium on Transportation Systems, Chania, Greece, June 1997.
5. Erzberger, H., "Design Principles and Algorithms for Automated Air Traffic Management," AGARD, Madrid, Spain, 6-7 November 1995.
6. Lee, K.K., & Davis, T.J., "The Development of the Final Approach Spacing Tool (FAST): A Cooperative Controller-Engineer Design Approach," 5th IFAC Symposium on Automated Systems Based on Human Skill, Berlin, Germany, September 1995.
7. Mueller, K.T., "Final Approach Spacing Tool (FAST) Velocity Accuracy Performance Analysis," Seagull Technology, Inc., 98148-3-01, April 1998.
8. Mueller, K.T., & Robinson, J.E. III, "Final Approach Spacing Tool (FAST) Velocity Accuracy Performance Analysis," American Institute of Aeronautics and Astronautics, Guidance, Navigation, and Control Conference, Boston, MA, 10-12 August 1998.
9. Papoulos, A., Probability, Random Variables, and Stochastic Processes, McGraw Hill, 1965.
10. Robinson III, J.E., Davis, T.J., & Isaacson, D.R., "Fuzzy Reasoning-Based Sequencing of Arrival Aircraft in the Terminal Area - Draft", NASA/Ames Research Center, April 1997.
11. Robinson III, J.E., Davis, T.J., & Isaacson, D.R., "Fuzzy Reasoning-Based Sequencing of Arrival Aircraft in the Terminal Area", AIAA Guidance, Navigation and Control Conference, New Orleans, LA, August 1997.
12. Slattery, R.A., "Terminal Area Trajectory Synthesis for Air Traffic Control Automation," Proceedings of the American Control Conference, Seattle, WA, June 1995.
13. Slattery, R.A., & Zhao, Y., "Trajectory Synthesis for Air Traffic Automation," Journal of Guidance, Control, and Dynamics, Vol. 20, No. 2, March-April 1997.

FAST Scheduling Logic Velocity Vector Sensitivity Analysis Appendices

Draft Document

K. Tysen Mueller

Prepared for:

National Aeronautics and Space Administration
Ames Research Center
Moffett Field, CA 94035-1000

Under Subcontract to:

Honeywell Inc.
Houston Engineering Center
Houston, TX 77058

Honeywell Subcontract No. 8021308
CLIN 006, Task Order 15
NASA Prime Contract No. NAS2-98001

Seagull Project No. C175.15

13 November 1998

APPENDIX A: FAST TS MATLAB SIMULATION

The FAST TS MATLAB simulation was developed and delivered under the previous study. It is included in this appendix because it is an integral part of the simulations developed and used under this study, and because it contains the default trajectories required under this study. In addition, it has been made more modular. For a complete discussion of the equations used in this MATLAB simulation, (Mueller, 1998) should be consulted.

A.1 Overview

The trajectory calculations are performed in fastTS.m. The listing for this MATLAB script is presented in Section A.3. The inputs are now contained in a separate MATLAB script called fastTSin.m. The listing for this script is found in Section A.4. Finally, the trajectory output is now plotted in a separate MATLAB function script called TSplot.m. Its listing is found in Section A.5.

The input routine contains the parameters for four default trajectories. In addition, it provides the structure for the user's own trajectory, if the default trajectories are not used. The assumptions and restrictions for defining a new trajectory are included in the comment statements of fastTSin.m.

The four default trajectory parameter sets are for a standard jet, a standard turboprop, a late time of arrival jet, and a late time of arrival turboprop trajectory. All four default trajectories use the respective jet or turboprop flight path segments from the southwest metering fix to Dallas-Ft. Worth Runway 18R. The late time of arrival trajectories also use the Base Extension flight path segment.

Since the main routine calculates each trajectory using two stages, the main routine currently prints to the screen a summary of each trajectory stage. The first stage derives the individual altitude, speed, and heading histories referenced to flight path distance traveled. These are merged to provide a distance-referenced trajectory. The second stage integrates the higher derivatives of this distance-referenced trajectory to obtain a time-referenced trajectory.

With the trajectory time history summary which is printed to the screen, key parameters can be identified. Specifically, the time it takes to reach the Downwind (null heading) segment is important. It is used to determine the FCTS when this same information is obtained for an aircraft merging onto the Downwind segment from another segment. Also the time and distances to reach the runway threshold can be obtained from these onscreen trajectory summaries.

A.2 Test Case

To illustrate the results which are obtained with fastTS.m, the case which is flagged in the fastTSin.m listing (Section A.4) was executed. The scenario is for a jet flying from the

southwest metering fix to Dallas-Ft. Worth Runway 18R. The jet trajectory was obtained by setting the default trajectory flag, TFLAG, to 1 in fastTSin.m.

The MATLAB screen capture for this run is shown in Table A-1. The first part is a distance-referenced summary of the trajectory. The distance interval is determined by events, such as the start or completion of a speed, altitude, or heading maneuver, or the completion of the trajectory. The second part of this screen capture shows the time-referenced summary of the trajectory for the same events as the first trajectory. By examining the heading history, the time when the Downwind trajectory segment is reached, can be identified. This is the first time at which a north (null) heading is reached.

The plots which are automatically generated for this case are presented in Figures A-1 through A-4. The first two plots were obtained from the distance-referenced trajectory calculation while the last two were obtained from the time-referenced trajectory calculations. This explains why Figures A-1 and A-3 are identical. They can be used to provide a consistency check for this case.

In addition, a sample of the output trajectory data file is presented in Table A-2. The format of this trajectory data file is summarized in Table A-3. As noted with Table A-3, the user may select either output unitflag option. However, the unitflag = 0 option must be selected if this data file is used as input into either fastSL1.m (Appendix B) or fastSL2.m (Appendix C).

Table A-1. fastTS.m Test Case MATLAB Screen Capture (TFLAG = 1)

ans =

| DIST, | HEAD, | ALT, | SPEED, | ACCEL, | HEAD RATE, | ALT RATE, | GLIDE SLOPE |
|-------|-------|-------|--------|---------|------------|-----------|-------------|
| (nm) | (deg) | (kft) | (kts) | (kts/s) | (deg/s) | (kft/s) | (deg) |

NTRAJ =

Columns 1 through 5

| | | | | |
|-------|--------|-------|--------|--------|
| 0 | 57.92 | 13.00 | 280.00 | 0 |
| 5.38 | 57.92 | 11.00 | 280.00 | - 1.07 |
| 7.45 | 57.92 | 11.00 | 250.00 | 0 |
| 12.02 | 57.92 | 11.00 | 250.00 | 0 |
| 12.05 | 57.92 | 10.99 | 250.00 | - 1.07 |
| 14.45 | 57.92 | 10.99 | 210.00 | 0 |
| 16.02 | 57.92 | 10.99 | 210.00 | 0 |
| 22.66 | 57.92 | 8.52 | 210.00 | 0 |
| 24.79 | 0 | 7.73 | 210.00 | 0 |
| 32.14 | 0 | 5.00 | 210.00 | 0 |
| 43.02 | 0 | 5.00 | 210.00 | 0 |
| 43.17 | 0 | 4.94 | 210.00 | 0 |
| 45.71 | 69.28 | 4.00 | 210.00 | 0 |
| 46.47 | 90.00 | 4.00 | 210.00 | 0 |
| 49.02 | 90.00 | 4.00 | 210.00 | 0 |
| 50.27 | 90.00 | 3.54 | 210.00 | 0 |
| 53.06 | 165.97 | 2.50 | 210.00 | - 1.07 |
| 53.57 | 180.00 | 2.50 | 200.37 | - 1.07 |
| 54.58 | 180.00 | 2.50 | 180.00 | 0 |
| 61.02 | 180.00 | 2.50 | 180.00 | 0 |

Columns 6 through 8

| | | |
|--------|--------|------|
| 0 | - 0.03 | 3.50 |
| 0 | 0 | 0 |
| 0 | 0 | 0 |
| 0 | - 0.03 | 3.50 |
| 0 | 0 | 0 |
| 0 | 0 | 0 |
| 0 | - 0.02 | 3.50 |
| - 1.59 | - 0.02 | 3.50 |
| 0 | - 0.02 | 3.50 |
| 0 | 0 | 0 |
| 0 | - 0.02 | 3.50 |
| 1.59 | - 0.02 | 3.50 |
| 1.59 | 0 | 0 |
| 0 | 0 | 0 |
| 0 | - 0.02 | 3.50 |
| 1.59 | - 0.02 | 3.50 |
| 1.59 | 0 | 0 |
| 0 | 0 | 0 |
| 0 | 0 | 0 |
| 0 | 0 | 0 |

Table A-1. fastTS.m Test Case MATLAB Screen Capture (TFLAG = 1) -- cont.

ans =

| TIME(sec) | DIST(nm) | ALT(kft) | HEAD(deg) | GRND SPEED(kts) |
|-----------|----------|----------|-----------|-----------------|
|-----------|----------|----------|-----------|-----------------|

FTRAJ =

| | | | | |
|---------|-------|-------|--------|--------|
| 0 | 0 | 13.00 | 57.92 | 280.00 |
| 69.00 | 5.37 | 11.01 | 57.92 | 280.00 |
| 97.00 | 7.43 | 11.00 | 57.92 | 250.34 |
| 163.00 | 12.01 | 11.00 | 57.92 | 250.00 |
| 163.50 | 12.05 | 10.99 | 57.92 | 250.00 |
| 201.00 | 14.44 | 10.99 | 57.92 | 210.03 |
| 228.00 | 16.02 | 10.99 | 57.92 | 210.00 |
| 341.50 | 22.64 | 8.53 | 57.92 | 210.00 |
| 378.00 | 24.77 | 7.74 | 0.44 | 210.00 |
| 504.00 | 32.12 | 5.01 | 0.00 | 210.00 |
| 690.50 | 43.00 | 5.00 | 0.00 | 210.00 |
| 693.00 | 43.14 | 4.95 | 0.00 | 210.00 |
| 737.00 | 45.71 | 4.00 | 69.21 | 210.00 |
| 750.00 | 46.47 | 4.00 | 89.88 | 210.00 |
| 793.50 | 49.01 | 4.00 | 90.00 | 210.00 |
| 815.00 | 50.26 | 3.54 | 90.00 | 210.00 |
| 862.50 | 53.03 | 2.51 | 165.23 | 210.00 |
| 871.50 | 53.55 | 2.50 | 179.54 | 200.90 |
| 891.00 | 54.58 | 2.50 | 180.00 | 180.11 |
| 1019.50 | 61.00 | 2.50 | 180.00 | 180.00 |

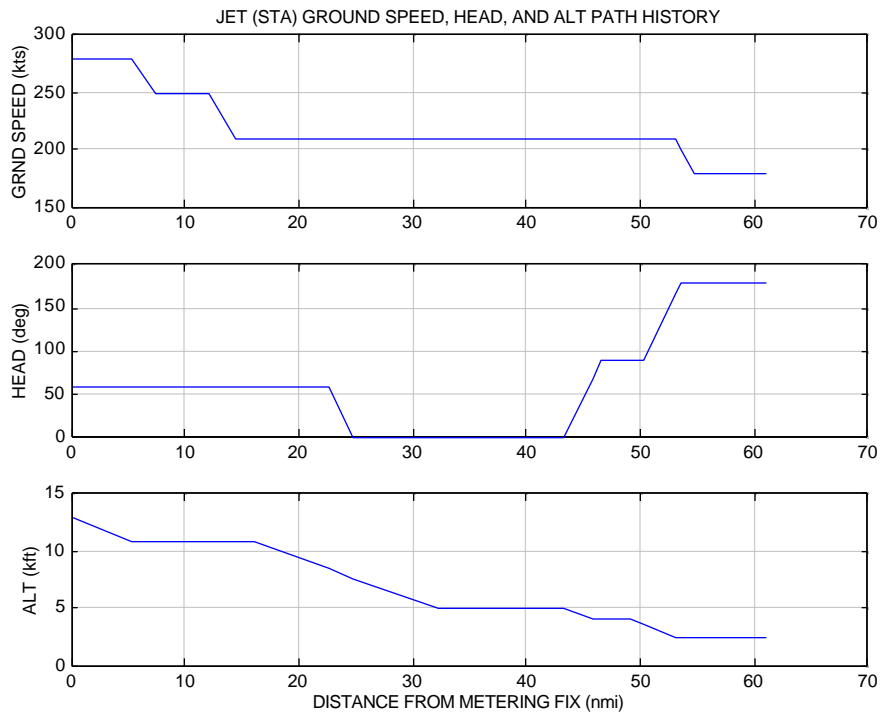


Figure A-1. fastTS.m Test Case Plot 1 (TFLAG = 1)

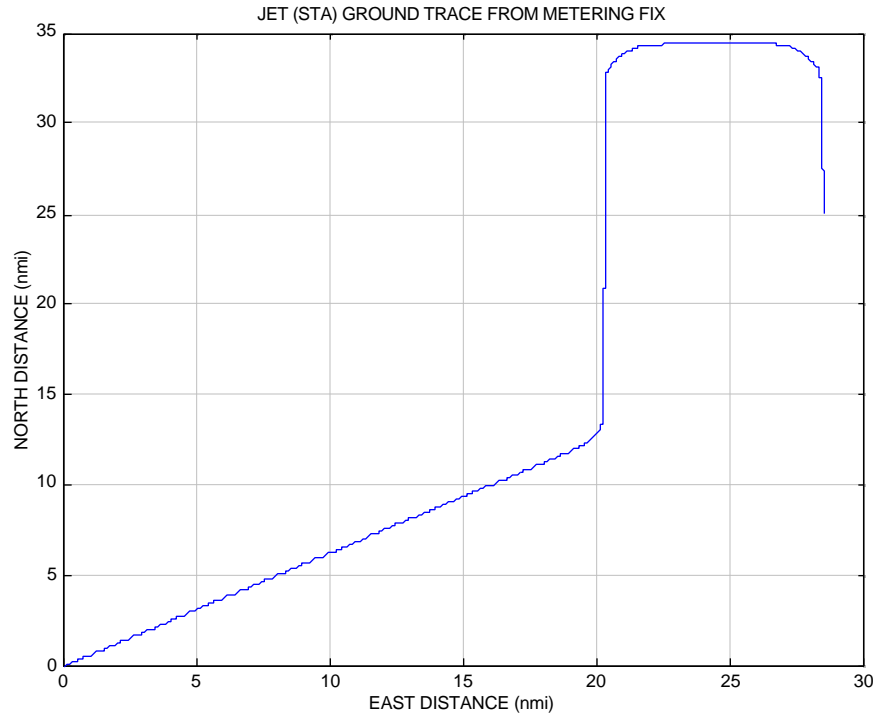


Figure A-2. fastTS.m Test Case Plot 2 (TFLAG = 1)

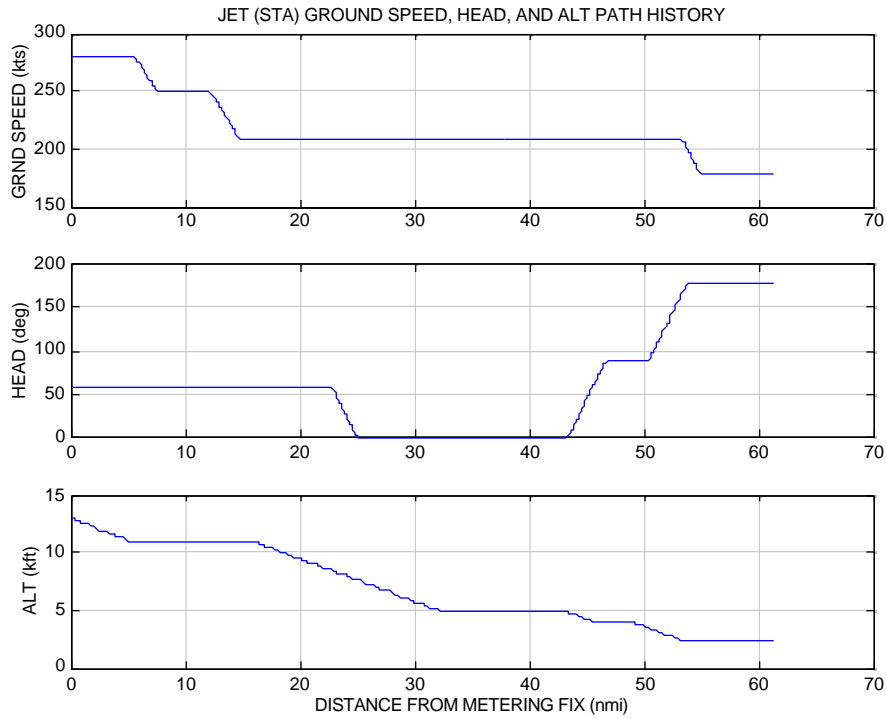


Figure A-3. fastTS.m Test Case Plot 3 (TFLAG = 1)

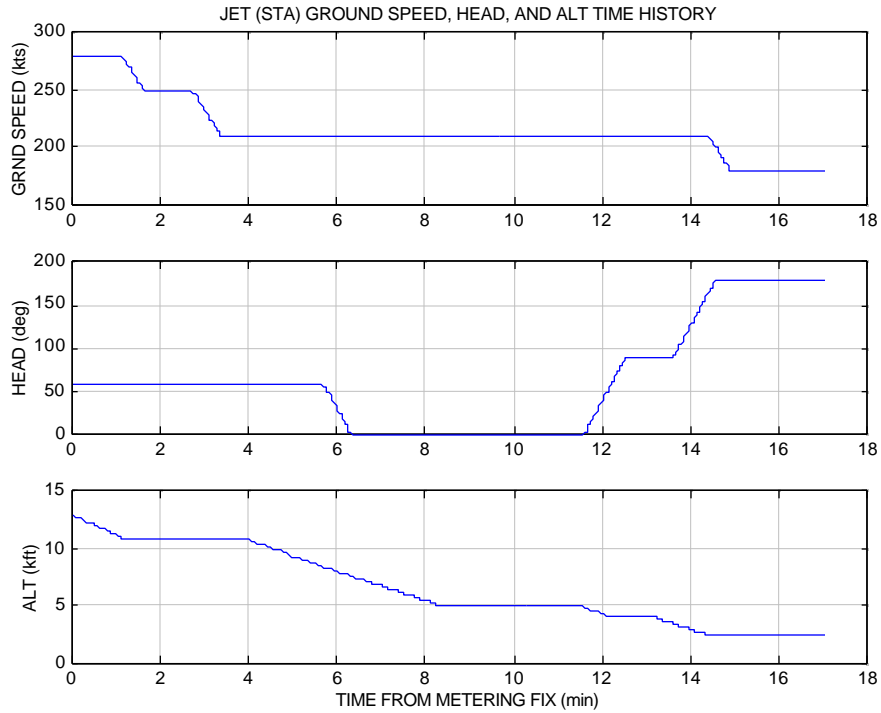


Figure A-4. fastTS.m Test Case Plot 4 (TFLAG = 1)

Table A-2. fastTSjet.dat Test Case Data File Abstract (unitflag = 0)

| | | | | | |
|----------------|---------------|---------------|---------------|---------------|----------------|
| 0.000000e+00 | 0.000000e+00 | 0.000000e+00 | 1.300000e+04 | 4.004158e+02 | 2.510069e+02 |
| -2.8904587e+01 | 0.000000e+00 | 5.7917813e+01 | 4.7258581e+02 | 0.000000e+00 | 0.000000e+00 |
| 5.000000e-01 | 2.0020793e+02 | 1.2550348e+02 | 1.2985548e+04 | 4.004158e+02 | 2.510069e+02 |
| -2.8904587e+01 | 2.3629291e+02 | 5.7917813e+01 | 4.7258581e+02 | 0.000000e+00 | 0.000000e+00 |
| 1.000000e+00 | 4.004158e+02 | 2.510069e+02 | 1.2971095e+04 | 4.004158e+02 | 2.510069e+02 |
| -2.8904587e+01 | 4.7258581e+02 | 5.7917813e+01 | 4.7258581e+02 | 0.000000e+00 | 0.000000e+00 |
| 1.500000e+00 | 6.0062379e+02 | 3.7651043e+02 | 1.2956643e+04 | 4.004158e+02 | 2.510069e+02 |
| -2.8904587e+01 | 7.0887872e+02 | 5.7917813e+01 | 4.7258581e+02 | 0.000000e+00 | 0.000000e+00 |
| 2.000000e+00 | 8.0083171e+02 | 5.0201391e+02 | 1.2942191e+04 | 4.004158e+02 | 2.510069e+02 |
| : | : | : | : | : | : |
| 1.018000e+03 | 1.7286314e+05 | 1.5268024e+05 | 2.500000e+03 | 3.8848741e+00 | -3.0174681e+02 |
| 0.000000e+00 | 3.7020332e+05 | 1.800000e+02 | 3.0380516e+02 | 0.000000e+00 | 0.000000e+00 |
| 1.018500e+03 | 1.7286508e+05 | 1.5252937e+05 | 2.500000e+03 | 3.8848741e+00 | -3.0174681e+02 |
| 0.000000e+00 | 3.7035522e+05 | 1.800000e+02 | 3.0380516e+02 | 0.000000e+00 | 0.000000e+00 |
| 1.019000e+03 | 1.7286702e+05 | 1.5237850e+05 | 2.500000e+03 | 3.8848741e+00 | -3.0174681e+02 |
| 0.000000e+00 | 3.7050712e+05 | 1.800000e+02 | 3.0380516e+02 | 0.000000e+00 | 0.000000e+00 |
| 1.019500e+03 | 1.7286896e+05 | 1.5222762e+05 | 2.500000e+03 | 3.8848741e+00 | -3.0174681e+02 |
| 0.000000e+00 | 3.7065903e+05 | 1.800000e+02 | 3.0380516e+02 | 0.000000e+00 | 0.000000e+00 |
| 1.020000e+03 | 1.7287091e+05 | 1.5207675e+05 | 2.500000e+03 | 3.8848741e+00 | -3.0174681e+02 |
| 0.000000e+00 | 3.7081093e+05 | 1.800000e+02 | 3.0380516e+02 | 0.000000e+00 | 0.000000e+00 |

Table A-3. fastTSjet.dat Test Case Data File Format

| Column | Variable | Units | |
|--------|---------------------|--------------------------|----------------|
| | | unitflag = 0 | unitflag = 1 |
| 1 | Time | seconds | seconds |
| 2 | East Position | feet | nautical miles |
| 3 | North Position | feet | nautical miles |
| 4 | Altitude | feet | kilofeet |
| 5 | East Velocity | feet/second | knots |
| 6 | North Velocity | feet/second | knots |
| 7 | Altitude Rate | feet/second | kilofeet/sec |
| 8 | Path Distance | feet | nautical miles |
| 9 | Heading | degrees | degrees |
| 10 | Ground Speed | feet/second | knots |
| 11 | Heading Rate | degrees/second | degrees/second |
| 12 | Ground Acceleration | feet/second ² | knots/second |

Note: fastTSjet.dat trajectory file must be generated with unitflag = 0 if input to fastSL1in.m or fastSL2in.m. unitflag does not affect units used for the screen capture trajectory summary.

A.3 fastTS.m Listing

```
% NASAfastTS.m
% FAST Trajectory Synthesizer calculations:
% Generates a simple TRACON TS trajectory
%
% Developed by:          K.Tysen Mueller
%                      SEAGULL TECHNOLOGY, INC.
% Date developed:       3 March 97
% Date updated:        15 October 98

clear all
format short e
t0 = clock;

%----- INPUT PARAMETERS -----

NASAfastTSin;          % Loads input parameters and flight path segment data base
% NOTE: There are four default trajectories in fastTSin -- user must
%       select which one or define their own

%----- INITIALIZATION -----

D2R = pi/180;          % deg to radians
NM2FT = 6076.1033;     % nautical miles to feet
HR2S = 3600;           % hours to seconds
```

```

KTS2FPS = NM2FT/HR2S;      % knots to feet/sec
KF2F = 1000;               % kilofeet to feet
g = 32.174;                % gravitational acceleration (ft/sec2)

%----- INITIAL TRAJECTORY CALCULATIONS -----

% It is assumed that altitude maneuvers are performed ahead of speed change maneuvers.
% Also, it is assumed that the altitude path - horizontal path profile provides a
% fixed trajectory which the aircraft will fly with a total time of flight dependent
% on the ground speed history.

% Compute nominal horizontal path history:
% Note that turning arcs are not considered except as finite heading changes.

totalpath = 0;

for i = 1:L-1
    xseg = (nodes(i+1,1) - nodes(i,1));
    yseg = (nodes(i+1,2) - nodes(i,2));

    seg(i,1) = sqrt(xseg^2 + yseg^2);
    seg(i,2) = atan2(xseg,yseg);

    if i == 1
        path(i,1) = seg(i,1);
    else
        path(i,1) = path(i-1,1) + seg(i,1);
    end
    path(i,2) = seg(i,2);
    totalpath = totalpath + seg(i,1);
end

% Adjust path for nodes to include finite heading changes:
% Assume that predominant speed will be the vgs ground speed, when computing
% the heading rate.

k = 0;
for i = 1:L-2
    k = k + 1;
    DPsi = path(i+1,2) - path(i,2);
    % (rads)
    if abs(DPsi) ~= 0
        Psidot = sign(DPsi)*(g*tan(phi*D2R))/(vgc*KTS2FPS); % (rads/sec)
        dtPsi = abs(DPsi/Psidot); % time to complete heading change (secs)
        dPath = (dtPsi/HR2S)*vgc; % distance to complete heading change (nmi)
        Path(k,1) = path(i,1) - (dPath/2); % Start heading change symmetrically prior
        Path(k,2) = path(i,2); % to node
        Path(k,3) = Psidot;
        Path(k+1,1) = path(i,1) + (dPath/2);
        Path(k+1,2) = path(i+1,2);
        Path(k+1,3) = 0;
        k = k + 1;
    else

```

```

        Path(k,1) = path(i,1);
        Path(k,2) = path(i,2);
        Path(k,3) = 0;
    end
end
k = k + 1;
    Path(k,1) = path(L-1,1);
    Path(k,2) = path(L-1,2);
    Path(k,3) = 0;
LL = k;

%PATH=Path

% Note that the path variable includes the cumulative horizontal distance
% traveled and the current heading.

% Compute nominal altitude profile history:

% Note that the hcontrol path distances, except the first one, is referenced
% relative to the outer marker. Also, changes in altitude occur with a theta
% glide slope. It is assumed that the aircraft is landing.

for i = 1:K
    if i == 1
        hseg(1,1) = 0;
        hseg(1,2) = h0;
        hseg(1,3) = -(KTS2FPS/KF2F)*Vg0*tan(hcontrol(1,3)*D2R);
        hseg(1,4) = hcontrol(1,3);
        hseg(2,1) = hseg(1,1) + (KF2F*(hseg(1,2) - hcontrol(1,2))/ ...
            (NM2FT*tan(hcontrol(1,3)*D2R)));
        hseg(2,2) = hcontrol(1,2);
        hseg(2,3) = 0;
        hseg(2,4) = 0;
    else
        hseg(2*i-1,1) = totalpath - hcontrol(i,1);
        hseg(2*i-1,2) = hseg(2*i-2,2);
        hseg(2*i-1,4) = hcontrol(i,3);
        if hseg(2*i-1,2) > vgcontrol(1,1)
            Vg = Vg0;
        elseif (hseg(2*i-1,2) <= vgcontrol(1,1)) & ...
            (hseg(2*i-1,2) > vgcontrol(2,1))
            Vg = vgcontrol(1,2);
        elseif hseg(2*i-1,2) <= vgcontrol(2,1)
            Vg = vgcontrol(2,2);
        end
        hseg(2*i-1,3) = -(KTS2FPS/KF2F)*Vg*tan(hcontrol(i,3)*D2R);
        hseg(2*i,1) = hseg(2*i-1,1) + ...
            (KF2F*(hseg(2*i-1,2) - hcontrol(i,2))/ ...
            (NM2FT*tan(hcontrol(i,3)*D2R)));
        hseg(2*i,2) = hcontrol(i,2);
        hseg(2*i,3) = 0;
        hseg(2*i,4) = 0;
    end
end
end

```

%HSEG=hseg

% Compute nominal ground speed history:

% Note that the speed changes are made after the nearest altitude change has
 % been completed. Also note that vgcontrol is referenced to a control altitude
 % while vgseg is referenced to distance traveled.

```
AG = (aG*HR2S)/(KTS2FPS);           % Speed change in units of : kts/HR
for i = 1:N
    for j = 1:K
        if i == 1
            if vgcontrol(1,1) == hseg(2*j,2)
                vgseg(1,1) = hseg(2*j,1);
                vgseg(1,2) = Vg0;
                vgseg(1,3) = AG/HR2S;
                vgseg(2,1) = vgseg(1,1) + ...
                    (vgcontrol(1,2)^2 - vgseg(1,2)^2)/(2*AG);
                vgseg(2,2) = vgcontrol(1,2);
                vgseg(2,3) = 0;
            end
        else
            if vgcontrol(i,1) == hseg(2*j,2) % Initiate Vg speed change after
                vgseg(2*i-1,1) = hseg(2*j,1); % control altitude has been reached
                vgseg(2*i-1,2) = vgseg(2*i-2,2);
                vgseg(2*i-1,3) = AG/HR2S;
                vgseg(2*i,1) = vgseg(2*i-1,1) + ...
                    (vgcontrol(i,2)^2 - vgseg(2*i-1,2)^2)/(2*AG);
                vgseg(2*i,2) = vgcontrol(i,2);
                vgseg(2*i,3) = 0;
            end
        end
    end
end
end
%VGSEG=vgseg
```

% Assemble nominal trajectory

% The nominal trajectory is a distance-referenced trajectory which has as its nodes
 % a change in the heading, altitude, or ground speed. Hence, it is not a fixed increment
 % distance history.

```
ntraj(1,1) = 0;           % distance (nmi)
ntraj(1,2) = Path(1,2);   % heading (rads)
ntraj(1,3) = h0;          % altitude (kft)
ntraj(1,4) = vgseg(1,2);  % ground speed (kts)
ntraj(1,5) = 0;           % ground accel (kts/s)
ntraj(1,6) = 0;           % heading rate (rads/s)
ntraj(1,7) = hseg(1,3);   % altitude rate (kft/s)
ntraj(1,8) = hseg(1,4);   % glide slope (deg)
```

```
for i = 1:LL
    ntraj(i+1,1) = Path(i,1);
    ntraj(i+1,2) = Path(i,2);
```

```

    ntraj(i+1,6) = Path(i,3);
end

for j = 1:2*K
    ntraj(j+LL+1,1) = hseg(j,1);
    ntraj(j+LL+1,3) = hseg(j,2);
    ntraj(j+LL+1,7) = hseg(j,3);
    ntraj(j+LL+1,8) = hseg(j,4);
end

for k = 1:2*N
    ntraj(k+LL+2*K+1,1) = vgseg(k,1);
    ntraj(k+LL+2*K+1,4) = vgseg(k,2);
    ntraj(k+LL+2*K+1,5) = vgseg(k,3);
end

% Search for and eliminate redundant distance entries:

% Assumes that some of the columns of two redundant rows will be identical
% or zero -- hence fill in each zero column from redundant non-zero column

k = 0;

for i = 1:(LL+2*K+2*N+1)
    if (ntraj(i,1) >= 0) % Check for previously flagged redundant rows
        k = k + 1;
        NT(k,:) = ntraj(i,:);
        if i < (LL+2*K+2*N+1)
            for j = (i+1):(LL+2*K+2*N+1)
                if (ntraj(i,1) == ntraj(j,1)) % Find redundant rows
                    for n = 2:8 % Merge redundant rows
                        if ntraj(i,n) == 0
                            NT(k,n) = ntraj(j,n);
                        end
                    end
                    ntraj(j,1) = -1; % Flag redundant row
                end
            end
        end
    end
end

KK = k;
%NT=NT

% Sort ntraj according to increasing distance:

[temp,I] = sort(NT);

for n=1:KK
    Ntraj(n,:) = NT(I(n,1),:);
end

% Fill in missing variables:

```

for i = 2:KK

```

    if Ntraj(i,4) == 0 % Fill in speeds
        if Ntraj(i-1,5) ~= 0 % Check if a speed change in progress
            dD = Ntraj(i,1) - Ntraj(i-1,1);
            dtVg = (-Ntraj(i-1,4) + sqrt((Ntraj(i-1,4))^2 + ...
                2*Ntraj(i-1,5)*dD*HR2S))/(Ntraj(i-1,5));
            Ntraj(i,4) = Ntraj(i-1,4) + dtVg*Ntraj(i-1,5);
            Ntraj(i,5) = Ntraj(i-1,5);
        else
            Ntraj(i,4) = Ntraj(i-1,4);
            Ntraj(i,5) = 0;
        end
    end

    if Ntraj(i,3) == 0 % Fill in altitudes
        if Ntraj(i-1,7) ~= 0 % Check if an altitude change in progress
            Ntraj(i,8) = Ntraj(i-1,8);
            Dh = -(Ntraj(i,1) - Ntraj(i-1,1))*(NM2FT/KF2F)*(tan(Ntraj(i-1,8)*D2R));
            Ntraj(i,3) = Ntraj(i-1,3) + Dh;
            Ntraj(i,7) = -Ntraj(i-1,4)*(KTS2FPS/KF2F)*(tan(Ntraj(i-1,8)*D2R));
        else
            Ntraj(i,3) = Ntraj(i-1,3);
            Ntraj(i,7) = 0;
            Ntraj(i,8) = 0;
        end
    end

    if Ntraj(i,2) == 0 % Fill in headings
        if Ntraj(i-1,6) ~= 0 % Check if a heading change in progress
            dtH = HR2S*(Ntraj(i,1) - Ntraj(i-1,1))/((Ntraj(i,4) + Ntraj(i-1,4))/2);
            Ntraj(i,2) = Ntraj(i-1,2) + Ntraj(i-1,6)*dtH;
            Ntraj(i,6) = Ntraj(i-1,6);
        else
            Ntraj(i,2) = Ntraj(i-1,2);
            Ntraj(i,6) = 0;
        end
    end

    if abs(Ntraj(i,2)) < 1e-4 % Eliminate any approximate null offsets
        Ntraj(i,2) = 0;
    end
end

```

% ----- FINAL TRAJECTORY CALCULATIONS -----

% The initial trajectory calculations yielded a distance-referenced history of discrete
 % variable changes. The final trajectory uses the initial trajectory to obtain a time
 % history of ENU position, velocity, and heading of the aircraft relative to the TRACON
 % radar location. The ENU coordinate system has its origin at the start of the trajectory
 % -- the metering gate.

% Calculate orthogonal ENU position and velocity history:

```
% t (sec) , x (ft) ,y (ft), z (ft), vx (ft/s), vy (ft/s), vz (ft/s), D (ft), psi (deg),
%      Vg(ft/s), psidot (deg/s), Vgdot (ft/s^2)
%
% The trajectory is obtained using a fixed time increment, trapezoidal integration scheme.
% This is modified to make sure that the specified control speeds, altitudes, and headings
% are reached, but not exceeded, by allowing the incremental changes to occur at time
% intervals less than the specified fixed incremental time interval. If a variable time
% history were selected, the incremental path would be used to determine the incremental
% time.

% Set up higher derivative variables for each segment

for k = 1:KK
    hdot(k) = Ntraj(k,7);
    psidot(k) = Ntraj(k,6);
    Vgdot(k) = Ntraj(k,5);
    theta(k) = Ntraj(k,8);
end

dtpsi = 0;
psidt = 0;
dtVg = 0;
Vgdt = 0;
k = 1;

for j = 1:tmax
    if k < KK
        ftraj(j,1) = (j - 1)*dt;          % Time (sec)

% Initialize trajectory:
        if j == 1
            Sx(1) = 0;                      % Accumulative east distance (nmi)
            Sy(1) = 0;                      % Accumulative north distance (nmi)
            Salt(1) = Ntraj(1,3);           % Accumulative altitude (kft)
            Svxx(1) = Ntraj(1,4)*sin(Ntraj(1,2)); % Accumulative east velocity (kts)
            Svyy(1) = Ntraj(1,4)*cos(Ntraj(1,2)); % Accumulative north velocity (kts)
            Spath(1) = 0;                   % Accumulative distance (nmi)
            Spsi(1) = Ntraj(1,2);           % Accumulative heading (rads)
            SVg(1) = Ntraj(1,4);            % Accumulative ground speed (kts)

% Compute hdot using current ground speed
            if hdot(1) ~= 0
                hdt = -SVg(1)*(KTS2FPS/KF2F)*(tan(theta(1)*D2R));
            else
                hdt = 0;
            end
            if unitflag == 0
                ftraj(1,2) = Sx(1)*NM2FT;    % East position (x - ft)
                ftraj(1,3) = Sy(1)*NM2FT;    % North position (y - ft)
                ftraj(1,4) = Salt(1)*KF2F;    % Altitude (h - ft)
                ftraj(1,5) = Svxx(1)*KTS2FPS; % East velocity (vx - ft/sec)
                ftraj(1,6) = Svyy(1)*KTS2FPS; % North velocity (vy - ft/sec)
            end
        end
    end
end
```

```

    ftraj(1,7) = hdt*KF2F;           % Altitude rate (ft/sec)
    ftraj(1,8) = Spath(1)*NM2FT;    % Distance (ft)
    ftraj(1,9) = Spsi(1)/D2R;       % Heading (deg)
    ftraj(1,10) = SVg(1)*KTS2FPS;   % Ground speed (ft/sec)
    ftraj(1,11) = psidot(1)/D2R;    % Heading rate (deg/sec)
    ftraj(1,12) = Vgdot(1)*KTS2FPS; % Ground acceleration (ft/sec^2)

% Set up time-referenced trajectory for printout
    FTRAJ(1,1) = ftraj(1,1); % Time (sec)
    FTRAJ(1,2) = Spath(1);   % Distance (nmi)
    FTRAJ(1,3) = Salt(1);    % Altitude (kft)
    FTRAJ(1,4) = ftraj(1,9); % Heading (deg)
    FTRAJ(1,5) = SVg(1);    % Speed (knots)
else
    ftraj(1,2) = Sx(1);
    ftraj(1,3) = Sy(1);
    ftraj(1,4) = Salt(1);
    ftraj(1,5) = Svx(1);
    ftraj(1,6) = Svy(1);
    ftraj(1,7) = hdt;
    ftraj(1,8) = Spath(1);
    ftraj(1,9) = Spsi(1)/D2R;
    ftraj(1,10) = SVg(1);
    ftraj(1,11) = psidot(1)/D2R;
    ftraj(1,12) = Vgdot(1);

% Set up time-referenced trajectory for printout
    FTRAJ(1,1) = ftraj(1,1); % Time (sec)
    FTRAJ(1,2) = ftraj(1,8); % Distance (nmi)
    FTRAJ(1,3) = ftraj(1,4); % Altitude (kft)
    FTRAJ(1,4) = ftraj(1,9); % Heading (deg)
    FTRAJ(1,5) = ftraj(1,10); % Speed (knots)
end
    jj = 1;

else

% Compute hdot using current ground speed
    if hdot(k) ~= 0
        hdt = -SVg(j-1)*(KTS2FPS/KF2F)*(tan(theta(k)*D2R));
    else
        hdt = 0;
    end
    if unitflag == 0
        ftraj(j,7) = hdt*KF2F;
        ftraj(j,11) = psidot(k)/D2R;
        ftraj(j,12) = Vgdot(k)*KTS2FPS;
    else
        ftraj(j,7) = hdt;
        ftraj(j,11) = psidot(k)/D2R;
        ftraj(j,12) = Vgdot(k);
    end
end

% Check to see if a control variable limit change will occur during

```

% this time/path interval

distance Dpath = SVg(j-1)*(dt/HR2S)+ 0.5*Vgdot(k)*(dt)*(dt/HR2S); % Incremental
Spath(j) = Spath(j-1) + Dpath;

if Spath(j) >= Ntraj(k+1,1) % Control variable change will
 dpath1 = (Ntraj(k+1,1) - Spath(j-1)); % occur during this interval

% Set up time-referenced trajectory for printout

 jj = jj+1;
 if unitflag == 0
 FTRAJ(jj,1) = ftraj(j-1,1); % Time (sec)
 FTRAJ(jj,2) = Spath(j-1); % Distance (nmi)
 FTRAJ(jj,3) = Salt(j-1); % Altitude (kft)
 FTRAJ(jj,4) = ftraj(j-1,9); % Heading (deg)
 FTRAJ(jj,5) = SVg(j-1); % Speed (knots)

 else
 FTRAJ(jj,1) = ftraj(j-1,1);
 FTRAJ(jj,2) = ftraj(j-1,8);
 FTRAJ(jj,3) = ftraj(j-1,4);
 FTRAJ(jj,4) = ftraj(j-1,9);
 FTRAJ(jj,5) = ftraj(j-1,10);
 end

 if Vgdot(k) == 0 % Compute time interval where control variables change
 dt1 = (dpath1/SVg(j-1))*HR2S;
 else
 dt1 = (-SVg(j-1)+sqrt((SVg(j-1))^2+2*Vgdot(k)*dpath1*HR2S))/Vgdot(k);
 end

 if dt1 > dt
 dt1 = dt;
 end

 dt2 = dt - dt1;

% Compute incremental trajectory changes:

% Incremental east distance
Dx1 = Sv_x(j-1)*(dt1/HR2S) + 0.5*(Vgdot(k)*sin(Spsi(j-1)) + ...
 SVg(j-1)*psidot(k)*cos(Spsi(j-1)))*(dt1)*(dt1/HR2S);
% Incremental north distance
Dy1 = Sv_y(j-1)*(dt1/HR2S) + 0.5*(Vgdot(k)*cos(Spsi(j-1)) - ...
 SVg(j-1)*psidot(k)*sin(Spsi(j-1)))*(dt1)*(dt1/HR2S);
Dalt1 = hdt*dt1; % Incremental altitude
Dvx1 = (Vgdot(k)*sin(Spsi(j-1)) + SVg(j-1)*psidot(k)*cos(Spsi(j-1)))*dt1;
Dvy1 = (Vgdot(k)*cos(Spsi(j-1)) - SVg(j-1)*psidot(k)*sin(Spsi(j-1)))*dt1;
% Incremental distance
Dpath1 = SVg(j-1)*(dt1/HR2S)+ 0.5*Vgdot(k)*(dt1)*(dt1/HR2S);
Dpsi1 = psidot(k)*dt1; % Incremental heading
Dvg1 = Vgdot(k)*dt1; % Incremental ground speed

Sx1 = Sx(j-1) + Dx1;
Sy1 = Sy(j-1) + Dy1;

```

Salt1 = Salt(j-1) + Dalt1;
Svx1 = Svx(j-1) + Dvx1;
Svy1 = Svy(j-1) + Dvy1;
Spath1 = Spath(j-1) + Dpath1;
SVg1 = SVg(j-1) + Dvg1;
Spsi1 = Spsi(j-1) + Dpsi1;

% Adjust incremental altitude not to exceed next control altitude --
% assuming descending aircraft.
    if Salt1 < Ntraj(k+1,3)
        Dalt1 = Ntraj(k+1,3) - Salt(j-1);
        Salt1 = Ntraj(k+1,3); % k'th control altitude reached
    end

% Adjust incremental heading not to exceed next control heading.
    if (Dpsi1 > 0) & (Spsi1 > Ntraj(k+1,2))
        Dpsi1 = Ntraj(k+1,2) - Spsi(j-1);
        Spsi1 = Ntraj(k+1,2); % k'th control heading reached
    elseif (Dpsi1 < 0) & (Spsi1 < Ntraj(k+1,2))
        Dpsi1 = Ntraj(k+1,2) - Spsi(j-1);
        Spsi1 = Ntraj(k+1,2);
    end

% Adjust incremental ground speed not to exceed next control ground speed --
% assuming aircraft is slowing down.
    if SVg1 < Ntraj(k+1,4)
        Dvg1 = Ntraj(k+1,4) - SVg(j-1);
        SVg1 = Ntraj(k+1,4); % k'th control ground speed reached
    end

% Compute hdot using current ground speed
    if hdot(k+1) ~= 0
        hdt = -SVg1*(KTS2FPS/KF2F)*(tan(theta(k+1)*D2R));
    else
        hdt = 0;
    end

    Dx2 = Svx1*(dt2/HR2S) + 0.5*(Vgdot(k+1)*sin(Spsi1) + ...
        SVg1*psidot(k+1)*cos(Spsi1))*(dt2)*(dt2/HR2S);
    Dy2 = Svy1*(dt2/HR2S) + 0.5*(Vgdot(k+1)*cos(Spsi1) - ...
        SVg1*psidot(k+1)*sin(Spsi1))*(dt2)*(dt2/HR2S);
    Dalt2 = hdt*dt2;
    Dvx2 = (Vgdot(k+1)*sin(Spsi1) + SVg1*psidot(k+1)*cos(Spsi1))*dt2;
    Dvy2 = (Vgdot(k+1)*cos(Spsi1) - SVg1*psidot(k+1)*sin(Spsi1))*dt2;
    Dpath2 = SVg1*(dt2/HR2S) + 0.5*Vgdot(k+1)*(dt2)*(dt2/HR2S);
    Dpsi2 = psidot(k+1)*dt2;
    Dvg2 = Vgdot(k+1)*dt2;

    Sx(j) = Sx1 + Dx2;
    Sy(j) = Sy1 + Dy2;
    Salt(j) = Salt1 + Dalt2;
    Svx(j) = Svx1 + Dvx2;
    Svy(j) = Svy1 + Dvy2;

```

```

    Spath(j) = Spath1 + Dpath2;
    SVg(j) = SVg1 + Dvg2;
    Spsi(j) = Spsi1 + Dpsi2;

    k = k + 1; % Segment is incremented based on accumulated distance

    if unitflag == 0
        ftraj(j,7) = hdt*KF2F;
        ftraj(j,11) = psidot(k)/D2R;
        ftraj(j,12) = Vgdot(k)*KTS2FPS;
    else
        ftraj(j,7) = hdt;
        ftraj(j,11) = psidot(k)/D2R;
        ftraj(j,12) = Vgdot(k);
    end
    else % No control variable changed during this time/path interval

% Compute incremental trajectory changes:
% Incremental east distance
    Dx = Svz(j-1)*(dt/HR2S) + 0.5*(Vgdot(k)*sin(Spsi(j-1)) + ...
        SVg(j-1)*psidot(k)*cos(Spsi(j-1)))*(dt)*(dt/HR2S);
% Incremental north distance
    Dy = Svy(j-1)*(dt/HR2S) + 0.5*(Vgdot(k)*cos(Spsi(j-1)) - ...
        SVg(j-1)*psidot(k)*sin(Spsi(j-1)))*(dt)*(dt/HR2S);
    Dalt = hdt*dt; % Incremental altitude
    Dvx = (Vgdot(k)*sin(Spsi(j-1)) + SVg(j-1)*psidot(k)*cos(Spsi(j-1)))*dt;
    Dvy = (Vgdot(k)*cos(Spsi(j-1)) - SVg(j-1)*psidot(k)*sin(Spsi(j-1)))*dt;
    Dpath = SVg(j-1)*(dt/HR2S) + 0.5*Vgdot(k)*(dt)*(dt/HR2S); % Incremental
distance
    Dpsi = psidot(k)*dt; % Incremental heading
    Dvg = Vgdot(k)*dt; % Incremental ground speed

    Sx(j) = Sx(j-1) + Dx;
    Sy(j) = Sy(j-1) + Dy;
    Salt(j) = Salt(j-1) + Dalt;
    Svz(j) = Svz(j-1) + Dvx;
    Svy(j) = Svy(j-1) + Dvy;
    Spath(j) = Spath(j-1) + Dpath;
    SVg(j) = SVg(j-1) + Dvg;
    Spsi(j) = Spsi(j-1) + Dpsi;

end % End of incremental path calculations

% Update trajectory:

if unitflag == 0
    ftraj(j,2) = Sx(j)*NM2FT;
    ftraj(j,3) = Sy(j)*NM2FT;
    ftraj(j,4) = Salt(j)*KF2F;
    ftraj(j,5) = Svz(j)*KTS2FPS;
    ftraj(j,6) = Svy(j)*KTS2FPS;

    ftraj(j,8) = Spath(j)*NM2FT;
    ftraj(j,9) = Spsi(j)/D2R;

```

```

        ftraj(j,10) = SVg(j)*KTS2FPS;
    else
        ftraj(j,2) = Sx(j);
        ftraj(j,3) = Sy(j);
        ftraj(j,4) = Salt(j);
        ftraj(j,5) = Svz(j);
        ftraj(j,6) = Svy(j);

        ftraj(j,8) = Spath(j);
        ftraj(j,9) = Spsi(j)/D2R;
        ftraj(j,10) = SVg(j);
    end
end
end
end
% End of: if j ~= 1 loop
% End of: if k < KK loop
% End of: j loop

% ----- SAVE & PLOT TRAJECTORY -----

NASATSplofunc(ptitle,Ntraj,ftraj);

% TFLAG = ?; 1 = Jet Scheduled Time of Arrival (STA) trajectory
%           2 = Turboprop STA trajectory
%           3 = Jet Late Time of Arrival (TA Late) trajectory
%           4 = Turboprop TA Late trajectory
%           5 = New (non-default) trajectory -- MUST BE DEFINED IN fastTSin.m

if TFLAG == 1

    save fastTSjet.dat    ftraj    -ascii

elseif TFLAG == 2

    save fastTSprop.dat   ftraj    -ascii

elseif TFLAG == 3

    save fastTSjetL.dat   ftraj    -ascii

elseif TFLAG == 4

    save fastTSpropL.dat  ftraj    -ascii

else

    save fastTSraj.dat    ftraj    -ascii

end

% Print out distance-referenced trajectory:

[ 'DIST, HEAD, ALT, SPEED, ACCEL, HEAD RATE, ALT RATE, GLIDE SLOPE'; ...
  '(nm) (deg) (kft) (kts) (kts/s) (deg/s) (kft/s) (deg) ' ]
format bank

```

```
NTRAJ = [Ntraj(:,1),Ntraj(:,2)/D2R,Ntraj(:,3),Ntraj(:,4),Ntraj(:,5),Ntraj(:,6)/D2R, ...
        Ntraj(:,7),Ntraj(:,8)];
NTRAJ = NTRAJ
```

% Print out time-referenced trajectory summary:

```
[' TIME(sec) DIST(nm) ALT(kft) HEAD(deg) GRND SPEED(kts)']
format bank
FTRAJ = FTRAJ
```

```
Run_time = etime(clock,t0)
```

A.4 fastTSin.m Listing

% **NASAFastTSin.m**

% Contains parameters required to run fastTS.m.

% Has 4 default aircraft trajectory parameter sets. All are for the SW metering fix

% approach to Dallas-Ft.Worth Runway 18R.

```
% Created by:          K.Tysen Mueller
%                      SEAGULL TECHNOLOGY, INC.
% Created on:          29 September 98
% Modified on:         16 October 98
```

```
%----- INPUT PARAMETERS -----
```

```
dt = 0.5;                % trajectory calculation time interval (sec)
Dt = 4.5;                % TRACON radar sweep interval -- must be integer
                        % multiple of dt (sec)
tmax = 5000;             % max number of time points
phi = 17;                % aircraft bank angle during turn (deg) -- T.Goka
aG = -1.8;               % aircraft acceleration during speed change (ft/sec2)
unitflag = 0;            % If non-zero, output trajectory will use non-standard
                        % units, such as: nmi, kft, knots, etc. Note: fastSLin.m,
                        % fastSL2in.m, fastSL.m, and fastSL2.m use standard units.
```

% Select one of four default trajectories or specify own:

```
TFLAG = 1; % 1 = Jet Scheduled Time of Arrival (STA) trajectory
          % 2 = Turboprop STA trajectory
          % 3 = Jet Late Time of Arrival (TA Late) trajectory
          % 4 = Turboprop TA Late trajectory
          % 5 = New (non-default) trajectory -- MUST BE DEFINED BELOW!
```

```
if TFLAG == 1
    K = 5;                % number of Jet STA control altitudes
    L = 5;                % number of Jet STA aircraft path nodes
    N = 3;                % number of Jet STA control speeds
    Vg0 = 280;            % initial Jet ground speed (kts)
```

```

h0 = 13;           % initial Jet altitude (kft)
ptitle = 'JET (STA)'; % plot title

elseif TFLAG == 2
    K = 5;           % number of Turbojet STA control altitudes
    L = 6;           % number of Turboprop STA aircraft path nodes
    N = 1;           % number of Turboprop STA control speeds
    Vg0 = 210;       % initial Turboprop ground speed (kts)
    h0 = 8;          % initial Turboprop aircraft altitude (kft)
    ptitle = 'TPROP (STA)'; % plot title

elseif TFLAG == 3
    K = 4;           % number of Jet TA Late control altitudes
    L = 5;           % number of Jet TA Late aircraft path nodes
    N = 2;           % number of Jet TA Late control speeds
    Vg0 = 280;       % initial Jet ground speed (kts)
    h0 = 13;         % initial Jet altitude (kft)
    ptitle = 'JET (TAL)'; % plot title

elseif TFLAG == 4
    K = 5;           % number of Turbojet TA Late control altitudes
    L = 6;           % number of Turboprop TA Late aircraft path nodes
    N = 1;           % number of Turboprop TA Late control speeds
    Vg0 = 210;       % initial Turboprop ground speed (kts)
    h0 = 8;          % initial Turboprop aircraft altitude (kft)
    ptitle = 'TPROP (TAL)'; % plot title

else
    % MUST BE SPECIFIED:
    K = 1;           % number of control altitudes
    L = 1;           % number of path nodes
    N = 1;           % number of control speeds
    Vg0 = 1;         % initial ground speed (kts)
    h0 = 1;          % initial altitude (kft)
    ptitle = "";      % plot title

end

% ----- FLIGHT PATH PARAMETER DATA BASE -----

% Four variables determine an aircraft flight path history: Altitude & Glide Slope
% (vs. Distance), Ground Speed (vs. Altitude), and East-North Locations of Turning
% Nodes. Turning nodes then determine the heading changes.

% Rules or Assumptions:

% 1) The trajectories always start at a metering fix -- hence first maneuver is always
%    an altitude maneuver.
% 2) Altitudes are referenced with respect to runway edge (or outer marker) except the
%    first one, which is referenced to metering fix.
% 3) Speed control maneuvers can only follow an altitude maneuver -- hence if a speed
%    maneuver is required, but there is no required altitude maneuver, add a small
%    altitude maneuver (e.g.: 10 ft).
% 4) Path nodes are specified in an east-north local level coordinate frame using any

```

% convenient origin (e.g: metering fix).
 % 5) Heading changes are determined by the change in orientation between two flight
 % path segments which are connected to the same node -- the turning node.
 % 6) Heading (Turning) rates are determined by the specified aircraft bank angle and
 % the predominant speed of the aircraft -- both must be specified.
 % 7) Altitude and heading changes may occur simultaneously, but speed changes cannot
 % be combined with any other maneuvers.
 % 8) Since a zero heading is used as a check for merging the separate distance derived
 % trajectories, an actual zero heading should be input as a small number (e.g: 0.01 deg).
 % 9) Scheduled Time of Arrival (STA) and Early Time of Arrival trajectories are assumed
 % to be nominally the same -- the former is obtained from the latter by adding a
 % minimum time separation spacing between two neighboring aircraft.
 % 10) Late Time of Arrival trajectories use a path extension and make most of their speed
 % reductions right after the first altitude maneuver.

```
hcontrol = zeros(K,3); nodes = zeros(L,2);    vgcontrol = zeros(N,2);
seg = zeros(L-1,2);    % ground path segments: length (nmi), azimuth (rads)
path = zeros(L-1,2); % ground path: cumulative distance (nmi), current azimuth (rads)
Path = zeros(2*L-3,3); % ground path: distance (nmi), azimuth (rads),
                        % azimuth rate (rads/s)
hseg = zeros(2*K,4); % altitude history: distance (nmi), altitude (kft),
                    % altitude rate (kft/s), glide slope (deg)
vgseg = zeros(2*N,3); % ground speed history: distance (nmi), speed (kts), accel (kts/s)
```

% Note: All default trajectories below are wrt Runway Edge

% ***** Altitude Control *****

% Altitude Control: distance from outer marker (nmi) -- except for initial distance,
 % altitude (kft), and glide slope (deg):

if TFLAG == 1

```
    hcontrol = [0 11.0 3.5; 49 10.99 3.5; 45 5.0 3.5; 18 4.0 3.5; 12 2.5 3.5];
```

elseif TFLAG == 2

```
    hcontrol = [0 7.99 3.5; 45 7.0 3.5; 35 5.0 3.5; 18 4.0 3.5; 12 2.5 3.5];
```

elseif TFLAG == 3

```
    hcontrol = [0 11.0 3.5; 76 5.0 3.5; 18 4.0 3.5; 12 2.5 3.5];
```

elseif TFLAG == 4

```
    hcontrol = [0 7.99 3.5; 76 7.0 3.5; 66 5.0 3.5; 18 4.0 3.5; 12 2.5 3.5];
```

else % MUST BE SPECIFIED:

```
    hcontrol = [1 1 1];
```

end

% ***** Ground Speed Control *****

% Ground Speed Control: altitude (kft) at which to start speed reduction, ground speed (kts).
% Revised 4/2/98 to incorporate conversion from CAS to ground speed.
% This is approximately: $V_g = CAS + 3 \text{ kts}/(\text{kft of altitude})$

if TFLAG == 1

vgcontrol = [11.0 250; 10.99 210; 2.5 180];
vgc = vgcontrol(2,2);

elseif TFLAG == 2

vgcontrol = [2.5 150];
vgc = Vg0;

elseif TFLAG == 3

vgcontrol = [11.0 210; 2.5 180];
vgc = vgcontrol(1,2);

elseif TFLAG == 4

vgcontrol = [2.5 150];
vgc = Vg0;

else % MUST BE SPECIFIED:

vgcontrol = [1 1];
vgc = 1;

end

% ***** Flight Path Segment Nodes *****

% Path Nodes: east pos. (nmi), north pos. (nmi), relative to metering gate.
% Note that small altitude changes may be introduced to permit speed changes.
% Also, note that one of nodes is slightly offset to avoid a null heading ...

if TFLAG == 1

nodes = [0 0; 20.1 12.6; 20.100001 33.7; 27.2 33.7; 27.2 24.6];

elseif TFLAG == 2

nodes = [0 0; 12.2 15.9; 20.1 23.0; 20.100001 33.7; 27.2 33.7; 27.2 24.6];

elseif TFLAG == 3

```
nodes = [0 0; 20.1 12.6; 20.100001 45.7; 27.2 45.7; 27.2 24.6];

elseif TFLAG == 4

    nodes = [0 0; 12.2 15.9; 20.1 23.0; 20.100001 45.7; 27.2 45.7; 27.2 24.6];

else    % MUST BE SPECIFIED:

    nodes = [1 1];

end
```

A.5 TSplotfunc.m Listing

```
% NASATsplotfunc.m
% Plots selected trajectory parameters from fastTS.m

% Created by:      K. Tysen Mueller
%                  SEAGULL TECHNOLOGY, INC.
% Created on:      29 September 98
% Modified on:     15 October 98

function[] = NASATsplotfunc(ptitle,Ntraj,fttraj)

D2R = pi/180;          % deg to radians
NM2FT = 6076.1033;     % nautical miles to feet
KTS2FPS = NM2FT/3600;  % knots to feet/sec
KF2F = 1000;          % kilofeet to feet

% Assuming that unitflag = 0:

% Distance-referenced trajectory plots:

subplot(3,1,1)
plot(Ntraj(:,1), Ntraj(:,4),'b')
title([ptitle,' GROUND SPEED, HEAD, AND ALT PATH HISTORY'])
ylabel('GRND SPEED (kts)','FontSize',10)
grid on

subplot(3,1,2)
plot(Ntraj(:,1), Ntraj(:,2)/D2R,'b')
ylabel('HEAD (deg)','FontSize',10)
grid on

subplot(3,1,3)
plot(Ntraj(:,1), Ntraj(:,3),'b')
xlabel('DISTANCE FROM METERING FIX (nmi)','FontSize',10)
ylabel('ALT (kft)','FontSize',10)
grid on
```

figure

% Time-referenced trajectory plots

```
plot(ftraj(:,2)/NM2FT, ftraj(:,3)/NM2FT,'b')
title([ptitle,' GROUND TRACE FROM METERING FIX'])
xlabel('EAST DISTANCE (nmi)','FontSize',10)
ylabel('NORTH DISTANCE (nmi)','FontSize',10)
grid on
```

figure

```
subplot(3,1,1)
plot(ftraj(:,8)/NM2FT, ftraj(:,10)/KTS2FPS,'b')
title([ptitle,' GROUND SPEED, HEAD, AND ALT PATH HISTORY'])
ylabel('GRND SPEED (kts)','FontSize',10)
grid on
```

```
subplot(3,1,2)
plot(ftraj(:,8)/NM2FT, ftraj(:,9),'b')
ylabel('HEAD (deg)','FontSize',10)
grid on
```

```
subplot(3,1,3)
plot(ftraj(:,8)/NM2FT, ftraj(:,4)/KF2F,'b')
xlabel('DISTANCE FROM METERING FIX (nmi)','FontSize',10)
ylabel('ALT (kft)','FontSize',10)
grid on
```

figure

```
subplot(3,1,1)
plot(ftraj(:,1)/60, ftraj(:,10)/KTS2FPS,'b')
title([ptitle,' GROUND SPEED, HEAD, AND ALT TIME HISTORY'])
ylabel('GRND SPEED (kts)','FontSize',10)
grid on
```

```
subplot(3,1,2)
plot(ftraj(:,1)/60, ftraj(:,9),'b')
ylabel('HEAD (deg)','FontSize',10)
grid on
```

```
subplot(3,1,3)
plot(ftraj(:,1)/60, ftraj(:,4)/KF2F,'b')
xlabel('TIME FROM METERING FIX (min)','FontSize',10)
ylabel('ALT (kft)','FontSize',10)
grid on
```

```
h=findobj('FontSize',10);    % Changes plot tick font size
set(h,'FontSize',10)
```

APPENDIX B ORDERING PROCEDURE PERFORMANCE SIMULATION

B.1 Overview

The FAST SL Ordering Procedure Performance Simulation is a MATLAB simulation which is called fastSL1.m. It is illustrated in Figure B-1 and its listing is presented in Section B.3. The principal inputs are two nominal aircraft trajectory files. These files may be generated using fastTS.m or can be obtained from some other source, so long as the fastTS.m trajectory format ASCII format is used.

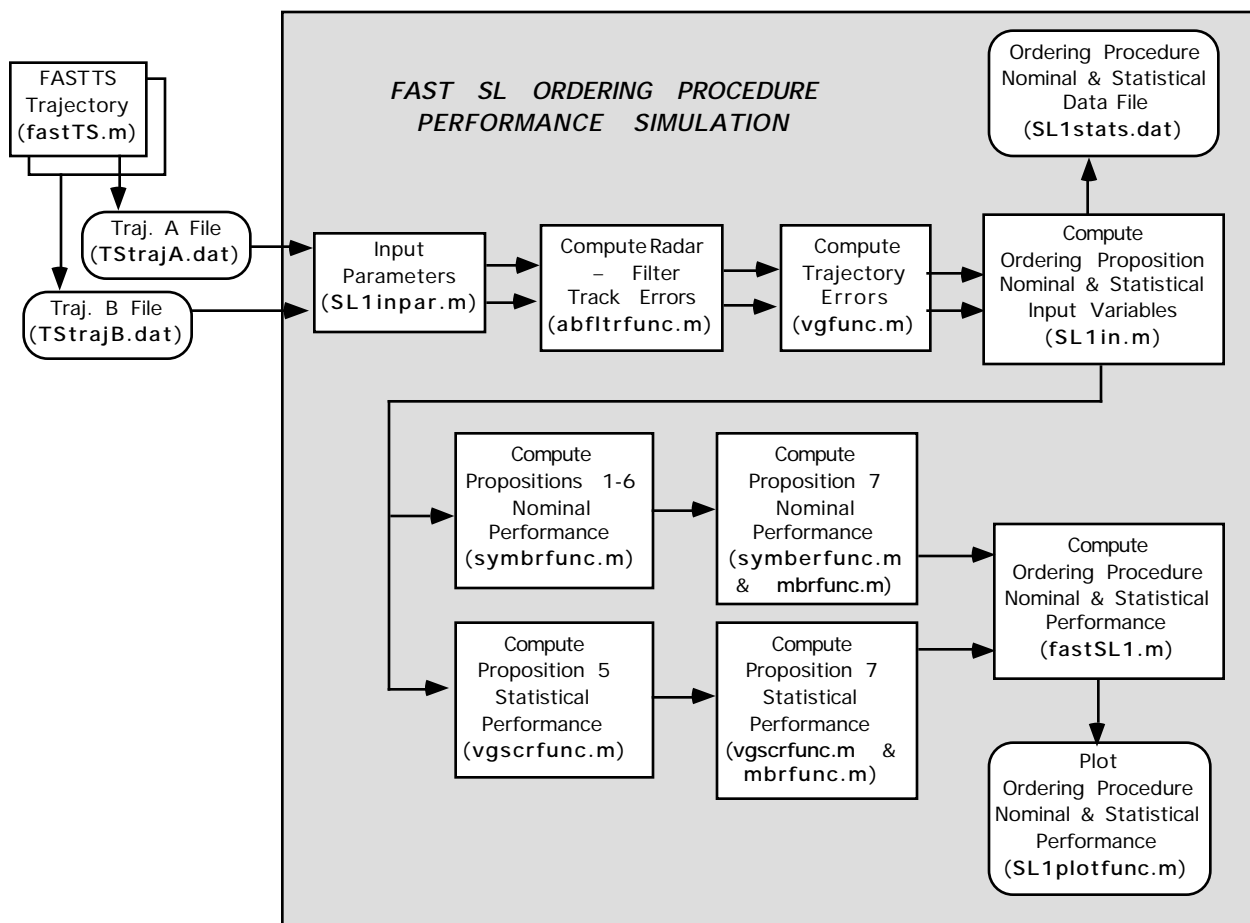


Figure B-1. FAST SL Ordering Procedure Performance Simulation

The simulation is divided into several major components. SL1inpar.m contains all the input parameters. It also includes trajectory time biases with which to establish the desired relative time or distance spacing between the two aircraft trajectories. A listing of SL1inpar.m is found in Section B.4.

Using the input parameters and the input trajectories, fastSL1in.m computes the trajectory tracking errors. It then computes from these tracking errors, the aircraft

relative nominal and statistical inputs required by FAST SL. The fastSL1in.m listing is found in Section B.5.

fastSL1in.m uses a number of MATLAB function scripts to compute these nominal and statistical relative input variables. These scripts include the abfltrfunc.m (Section B.6) and the vgfunc.m (Section B.7). abfltrfunc.m computes the radar tracking errors for the radar – tracking filter. vgfunc.m uses the outputs from the last two functions to compute the trajectory errors. Finally, fastSL1in.m uses the resulting trajectory errors and the nominal trajectory variables to compute the nominal and statistical relative input variables for the main routine, fastSL1.m.

Using these nominal and statistical input variables, fastSL1.m calls symbrfunc.m (Section B.8) and mbrfunc.m (Section B.10) to compute the nominal Proposition Membership, Output, Weight, and Firing Strength. fastSL1.m also calls vgscrfunc.m (Section B.9) to compute the statistical Proposition Membership, Output, Weight, and Firing Strength. Based on the nominal and statistical Proposition results, fastSL1.m computes the Ordering Procedure nominal and statistical Output, Weight, and Firing Strength. The Proposition and Procedure results are then plotted in SL1plotfunc.m (Section B.11).

B.2 Test Case

To execute fastSL1.m, the input parameter script, fastSL1inpar.m must be consulted and updated for the specific case to be executed. Also, it is recommended that SL1plotfunc.m is consulted to determine whether the plot axis assignments are current. For a new run, it is recommended that these axis assignments are commented out by using the MATLAB global replace function. This will avoid having the results be plotted off screen.

fastSL1.m can now be executed (Type: fastSL1 in the Command Window of MATLAB). For the specific fastSL1inpar.m set listed in Section B.4, plots B-2 through B-12 are automatically generated. The first two plots are based on parameters computed in fastSLin.m, while the remaining ten plots are obtained from the calculations performed directly in fastSL1.m.

In addition, a time history data file of the nominal and statistical input variables required by fastSL1.m is generated. Table B-1 presents an abstract of this file, called fastSL1stats.dat, while Table B-2 summarizes the format.

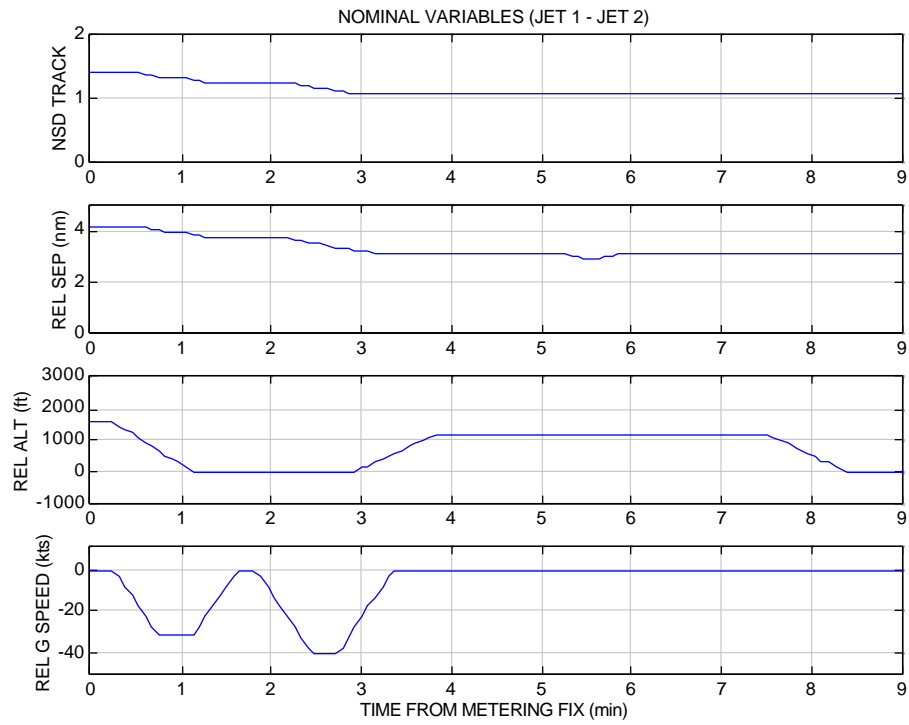


Figure B-2. Nominal Relative Variables (Jet 1 - Jet 2)

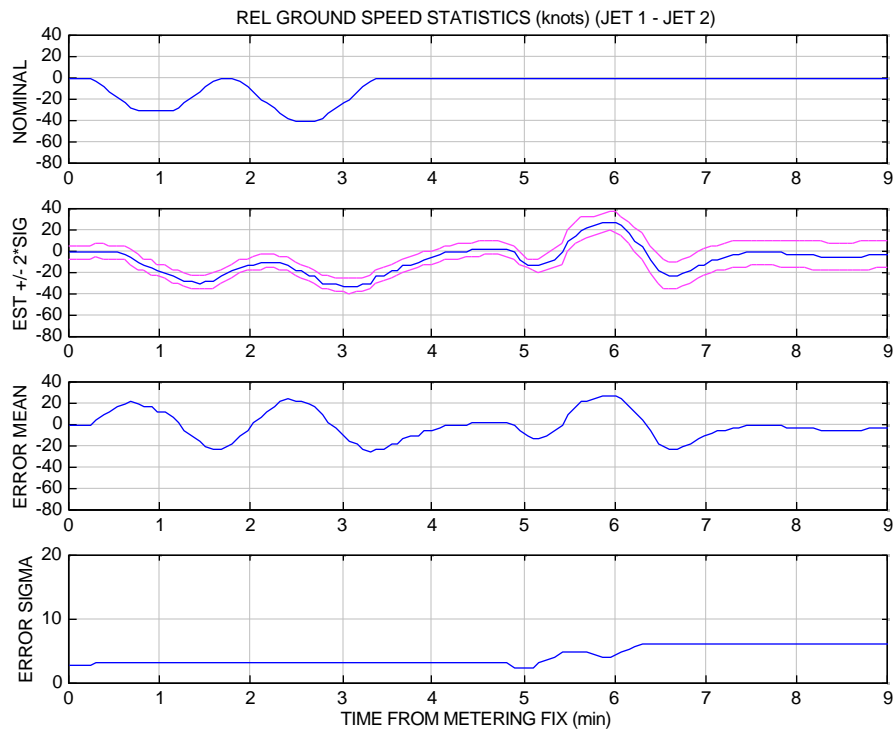


Figure B-3. Relative Ground Speed Statistics (Jet 1 - Jet 2)

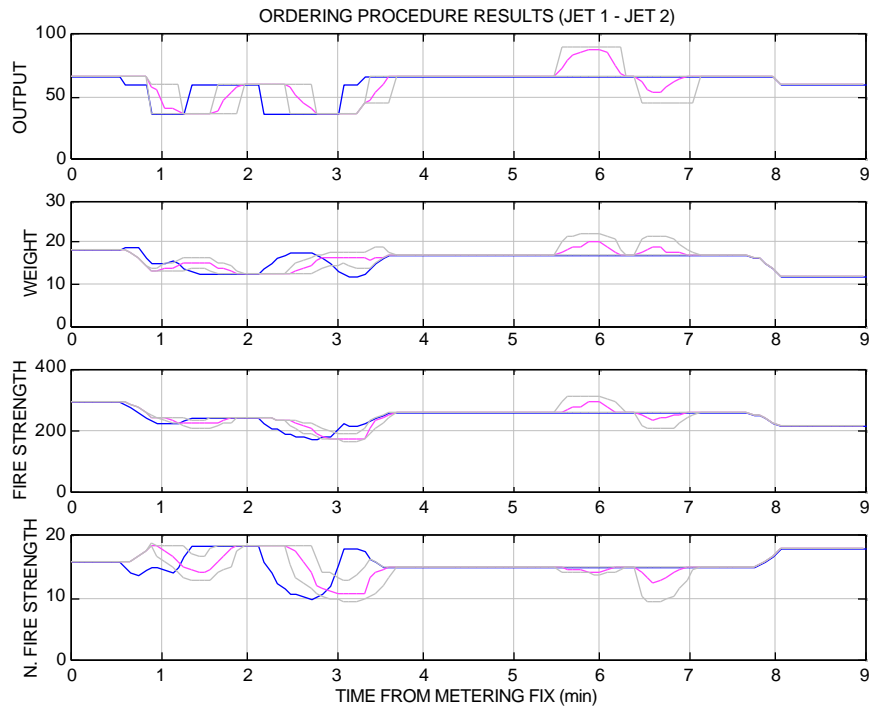


Figure B-4. Ordering Procedure Nominal and Statistical Results (Jet 1 - Jet 2)

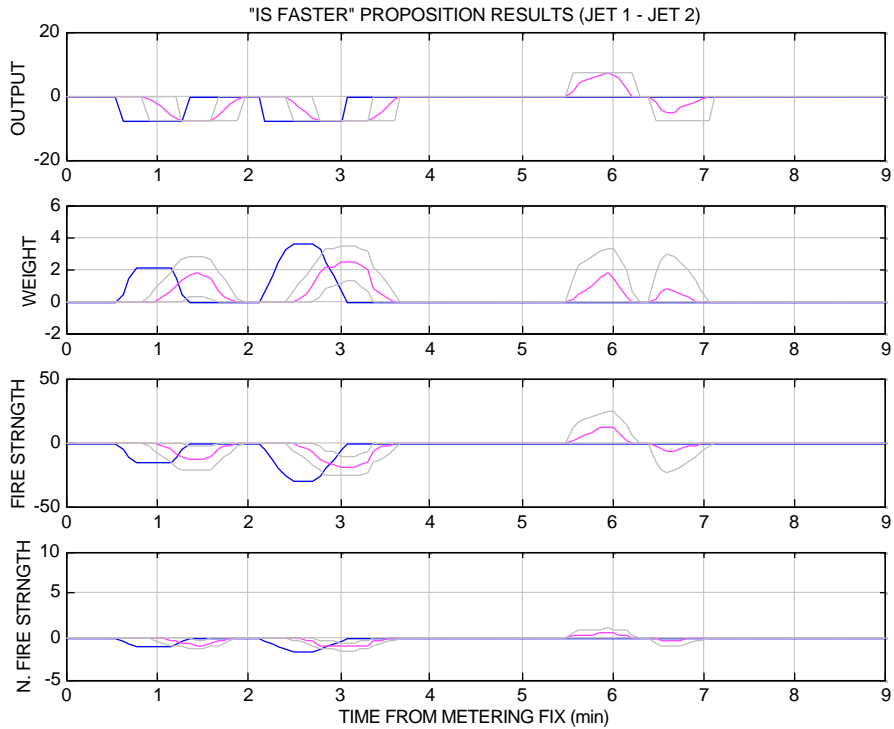


Figure B-5. 'Is Faster' Proposition Nominal and Statistical Results (Jet 1 - Jet 2)

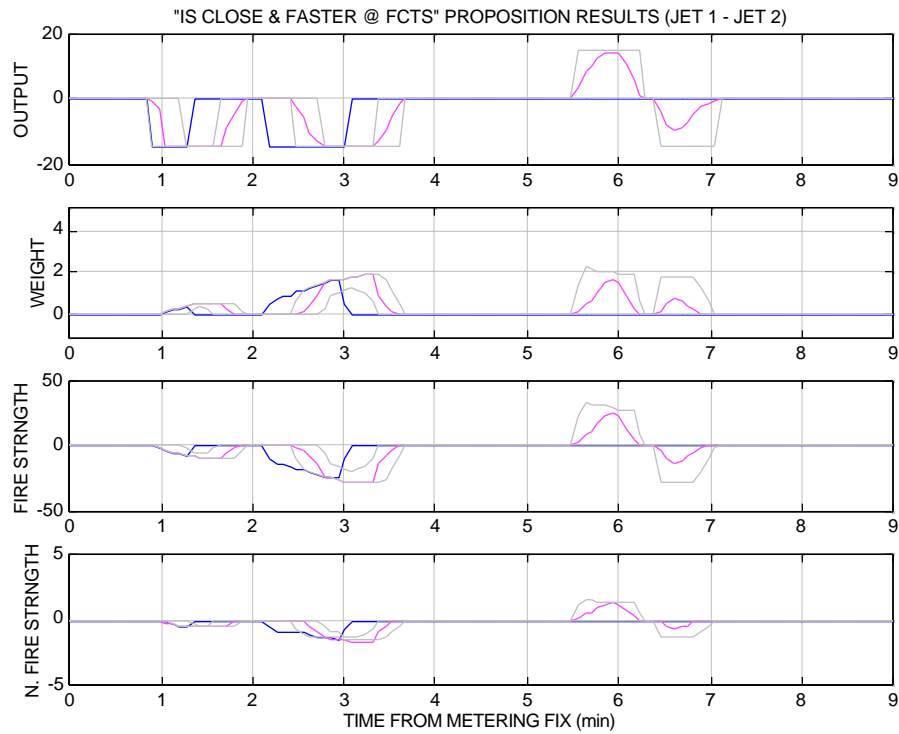


Figure B-6. 'Is Closer and Faster' Proposition Nominal and Statistical Results (Jet 1 - Jet 2)

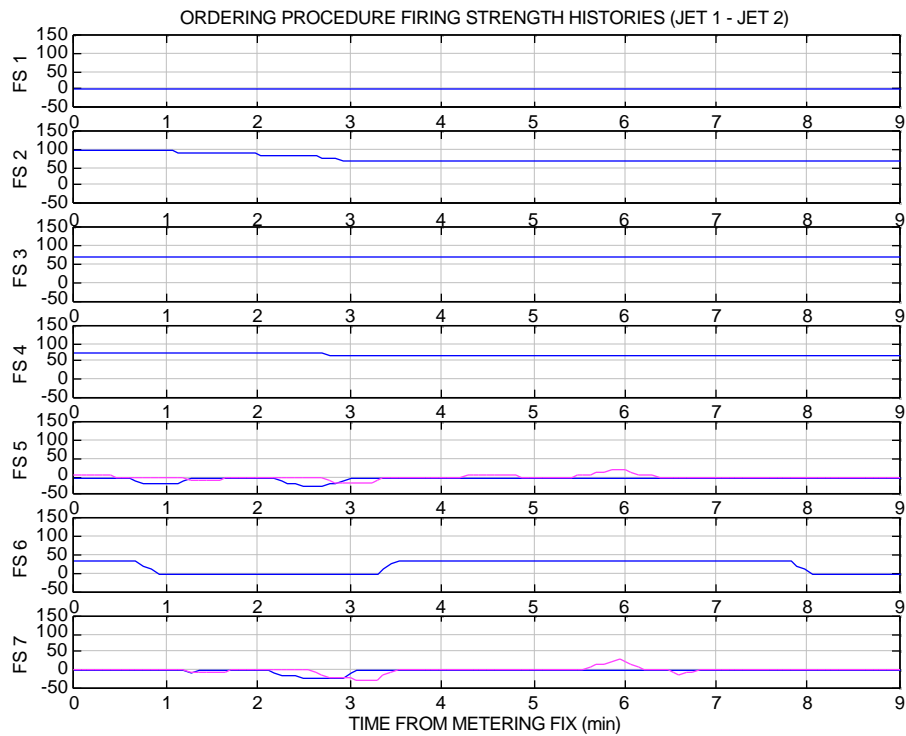


Figure B-7. Proposition Nominal and Statistical Firing Strength Results (Jet 1 - Jet 2)

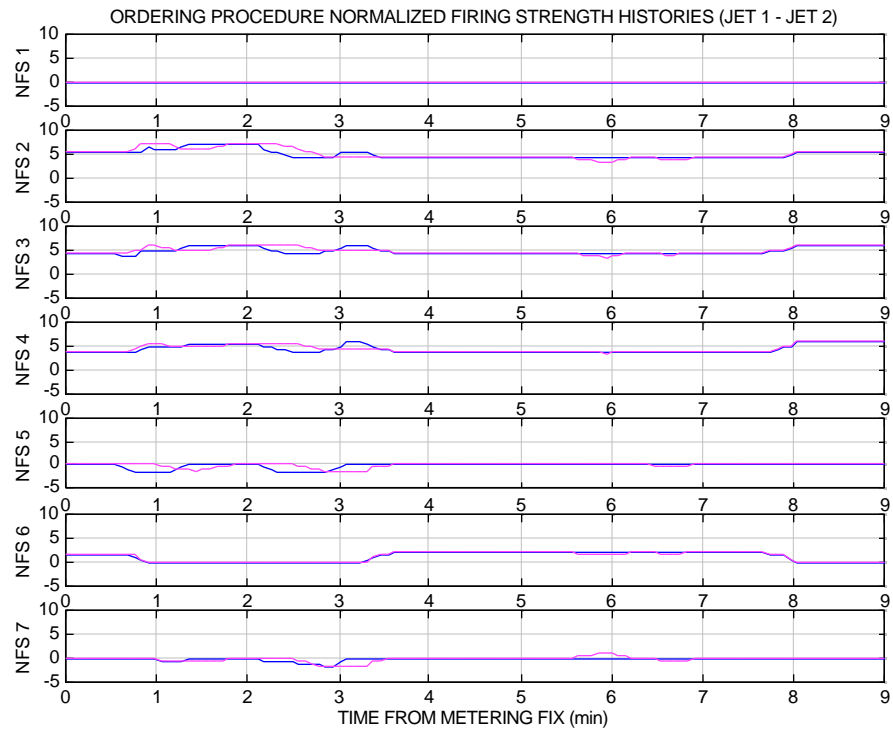


Figure B-8. Proposition Nominal and Statistical Normalized Firing Strength Results (Jet 1 - Jet 2)

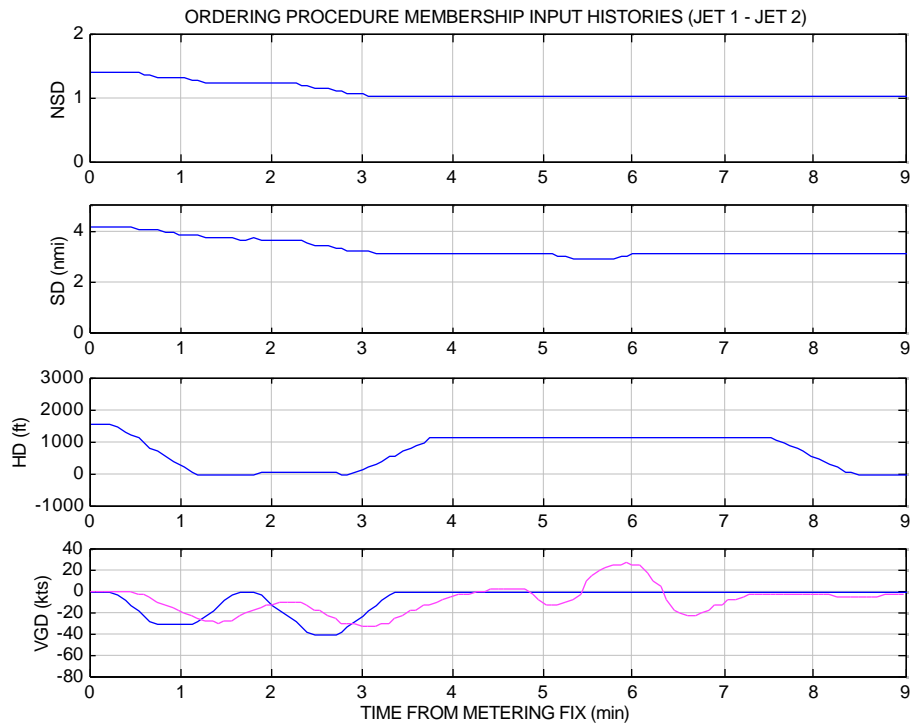


Figure B-9. Proposition Nominal and Statistical Input Variables (Jet 1 - Jet 2)

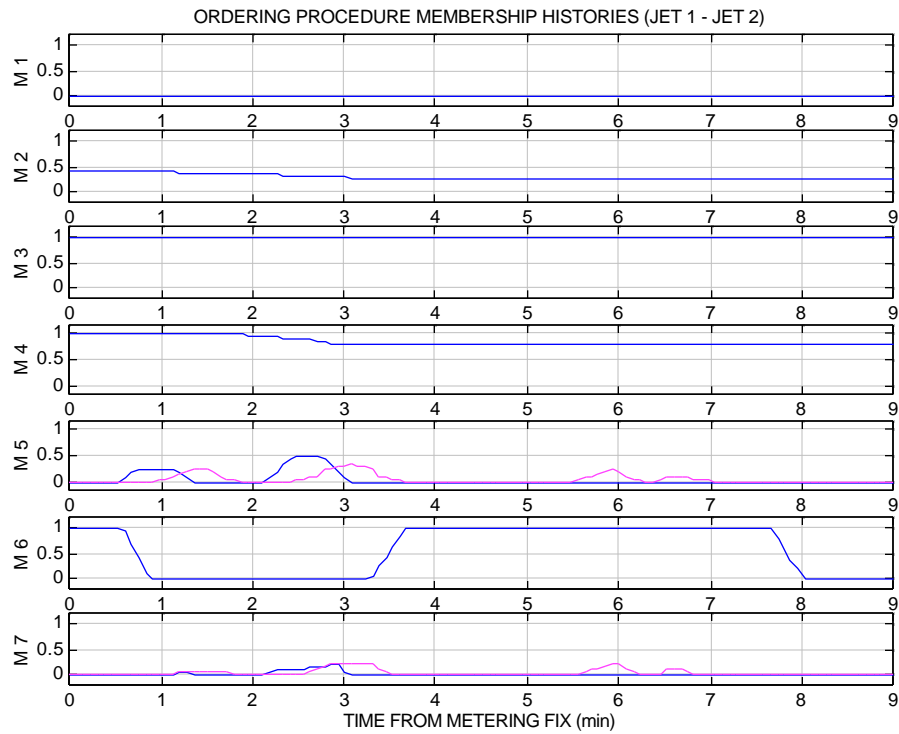


Figure B-10. Proposition Nominal and Statistical Membership Values (Jet 1 - Jet 2)

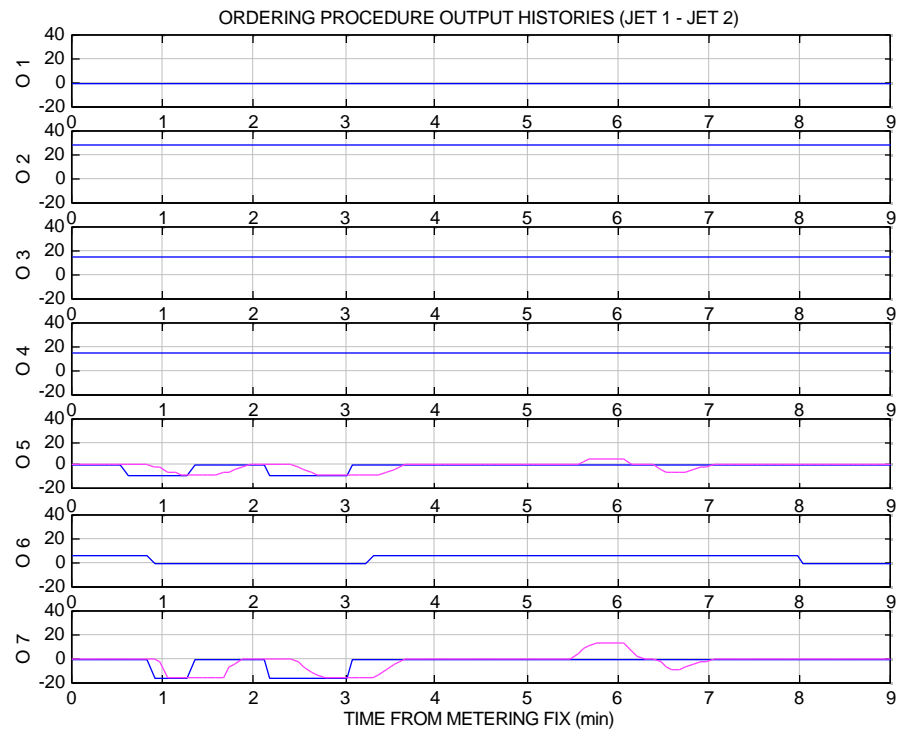


Figure B-11. Proposition Nominal and Statistical Output Values (Jet 1 - Jet 2)

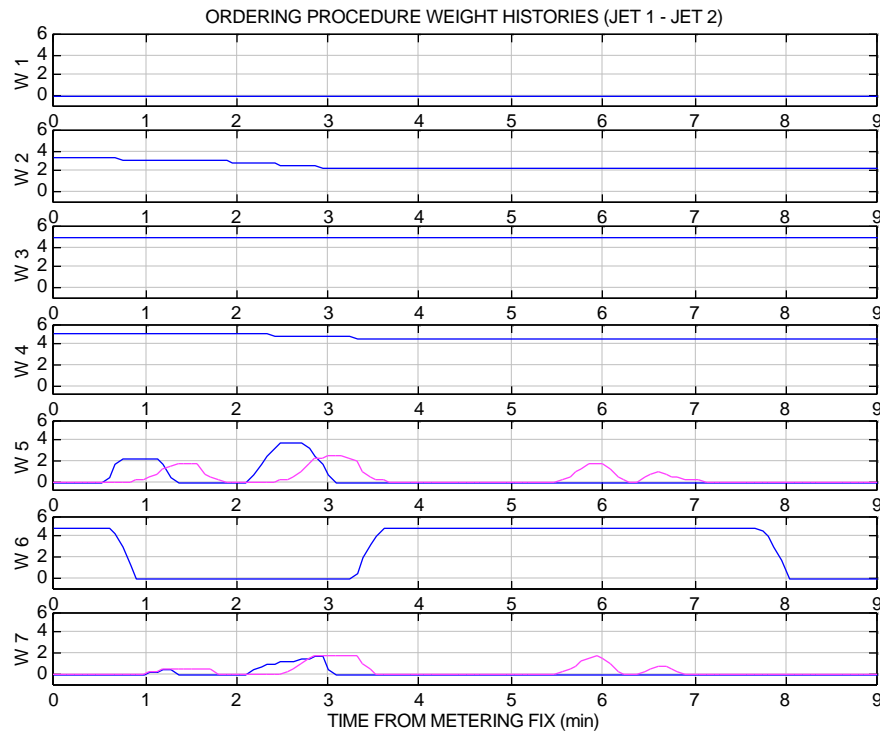


Figure B-12. Proposition Nominal and Statistical Weight (Jet 1 - Jet 2)

Table B-1. fastSL1stats.dat Statistics Data File Abstract

| | | | | | |
|---------------|---------------|---------------|--------------|---------------|----------------|
| 0.000000e+00 | 1.400000e+00 | 4.200000e+00 | 1.560848e+03 | 0.000000e+00 | -1.1686135e-01 |
| 2.8810623e+00 | | | | | |
| 7.500000e-02 | 1.400000e+00 | 4.200000e+00 | 1.560847e+03 | 0.000000e+00 | -1.3805091e-01 |
| 2.9327011e+00 | | | | | |
| 1.500000e-01 | 1.400000e+00 | 4.200000e+00 | 1.560848e+03 | 0.000000e+00 | -1.4372000e-01 |
| 2.9422638e+00 | | | | | |
| 2.250000e-01 | 1.400000e+00 | 4.199999e+00 | 1.560848e+03 | 0.000000e+00 | -1.3696301e-01 |
| 2.9862068e+00 | | | | | |
| 3.000000e-01 | 1.399611e+00 | 4.198833e+00 | 1.479717e+03 | -2.993406e+00 | -3.8767509e+00 |
| 3.0369844e+00 | | | | | |
| : | : | : | : | : | : |
| 8.700000e+00 | 1.0499997e+00 | 3.1684008e+00 | 0.000000e+00 | 0.000000e+00 | 3.8803258e+00 |
| 6.1116641e+00 | | | | | |
| 8.775000e+00 | 1.0500003e+00 | 3.1684024e+00 | 0.000000e+00 | 0.000000e+00 | 3.7045629e+00 |
| 6.1112088e+00 | | | | | |
| 8.850000e+00 | 1.0500003e+00 | 3.1684008e+00 | 0.000000e+00 | 0.000000e+00 | 3.4646246e+00 |
| 6.1107361e+00 | | | | | |
| 8.925000e+00 | 1.0499997e+00 | 3.1684008e+00 | 0.000000e+00 | 0.000000e+00 | 3.1675372e+00 |
| 6.1102443e+00 | | | | | |
| 9.000000e+00 | 1.0499997e+00 | 3.1684024e+00 | 0.000000e+00 | 0.000000e+00 | 2.821372e+00 |
| 6.1097331e+00 | | | | | |

Table B-2. fastSL1stats.dat File Format

| COLUMN | VARIABLE | UNITS |
|--------|---|-------|
| 1 | Time | min |
| 2 | Nominal NSD Track | |
| 3 | Nominal Relative Separation | nm |
| 4 | Nominal Relative Altitude | ft |
| 5 | Nominal Relative Ground Speed | kts |
| 6 | Mean Relative Ground Speed | kts |
| 7 | Standard Deviation of Relative Ground Speed | kts |

B.3 fastSL1.m MATLAB Listing

```
% NASAFastSL1.m
% Simulation of FAST SL nominal fuzzy decision logic. It takes the ENU
% time-referenced aircraft trajectory for two aircraft following different
% paths to the same runway. By considering these as generic trajectories, multiple
% aircraft are simulated by time shifting the two input trajectories. The nominal
% trajectory variables are then used to evaluate the:
%
%      Ordering Procedure of a GENERAL-Type Spatial Constraint
% -----

% developed by:          K. Tysen Mueller
%                      SEAGULL TECHNOLOGY, INC.
% developed on:          1 October 98
% modified on:           9 November 98

clear all
t0 = clock;

% ----- INPUTS -----

NASAFastSL1inpar; % Contains control constants, data, and trajectory files
                % NOTE: User must start with this file before executing fasSL1in.m

NASAFastSL1in;      % Simulation of FAST SL input errors based on the alpha-beta filter
errors.
                % It takes the ENU time-referenced aircraft trajectory and the radar error
% statistics and computes the time-referenced radar, ENU, time of arrival (ETA), and First
% Common Time Step (FCTS) estimation error mean and sigmas. Based on original
% alphabeta.m and on fastSLin.m simulations. Provides inputs for fastSL1.m.

%----- INITIALIZATION -----

D2R = pi/180;          % deg to radians
NM2FT = 6076.1033;     % nautical miles to feet
HR2S = 3600;           % hours to seconds
KTS2FPS = NM2FT/HR2S; % knots to feet/sec
```

Seagull Technology, Inc. Proprietary Information

KF2F = 1000;
M2S = 60;

% kilofeet to feet
% Minutes to seconds

% ----- MAIN LOOP -----

for k = 1:K

% ***** ORDERING PROCEDURE of a GENERAL-Type (OPGT) Spatial Constraint *****

% OPGT propositions:

% O(k): Proposition output value

% M(k) : Proposition membership

% W(k) : Proposition weight

% S(k) : Proposition firing strength (weighted output)

% 'Significantly Ahead' Membership, 'Significantly Favored' Consequent Functions:

[O1(k),M1(k),W1(k),S1(k)] = NASAsymbrfunc(NSD(k),2,4,45);

% 'Ahead' Membership, 'Favored' Consequent Functions:

[O2(k),M2(k),W2(k),S2(k)] = NASAsymbrfunc(NSD(k),0.5,2.5,30);

% 'Slightly Ahead' Membership, 'Slightly Favored' Consequent Functions:

[O3(k),M3(k),W3(k),S3(k)] = NASAsymbrfunc(NSD(k),0,1,15);

% 'Ahead at Current Position' Membership, 'Slightly Favored' Consequent Functions:

[O4(k),M4(k),W4(k),S4(k)] = NASAsymbrfunc(NSD(k),0.25,1.25,15);

% 'Faster' Membership, 'Marginally Favored' Consequent Functions:

[O5(k),M5(k),W5(k),S5(k)] = NASAsymbrfunc(dVG(k),20,60,7.5); % Nominal

[sO5(k),sM5(k),sW5(k),sS5(k)] = NASAvgscrfunc(dVGest(k),sigdVG(k),20,60,7.5); %
Statistical

[sO5a(k),sM5a(k),sW5a(k),sS5a(k)] = NASAsymbrfunc(dVGestM2S(k),20,60,7.5); % Min

[sO5b(k),sM5b(k),sW5b(k),sS5b(k)] = NASAsymbrfunc(dVGestP2S(k),20,60,7.5); % Max

% 'Lower at Current Position' Membership, 'Marginally Favored' Consequent Functions:

[O6(k),M6(k),W6(k),S6(k)] = NASAsymbrfunc(dh(k),500,1000,7.5);

% Note that Proposition 7 uses an AND operation for the 'Close' and 'Faster'
% membership functions. This AND operation selects the output from these two
% membership functions for which there is minimum membership.

% 'Close at Current Position' Membership, 'Slightly Favored' Consequent Functions:

[Out,Mbr,Wght,FS] = NASAmbrfunc(ds(k),0,1,4,0,15);

M7(k) = 0; O7(k) = 0; W7(k) = 0; S7(k) = 0;

% Nominal

[O7v(k),M7v(k),W7v(k),S7v(k)] = NASAasybrfunc(dVG(k),20,60,15); % Nominal

if M7v(k) >= Mbr

M7(k) = Mbr;
Sgn = sign(dVG(k));
O7(k) = Out*Sgn;
W7(k) = Wght;
S7(k) = FS*Sgn;

elseif M7v(k) < Mbr

M7(k) = M7v(k);
O7(k) = O7v(k);
W7(k) = W7v(k);
S7(k) = S7v(k);

end

% Statistical

sM7(k) = 0; sO7(k) = 0; sW7(k) = 0; sS7(k) = 0;

[sO7v(k),sM7v(k),sW7v(k),sS7v(k)] = NASAvgscrfunc(dVGest(k),sigdVG(k),20,60,15); %
Statistical

if sM7v(k) >= Mbr

sM7(k) = Mbr;
sSgn = sign(dVGest(k));
sO7(k) = Out*sSgn;
sW7(k) = Wght;
sS7(k) = FS*sSgn;

elseif sM7v(k) < Mbr

sM7(k) = sM7v(k);
sO7(k) = sO7v(k);
sW7(k) = sW7v(k);
sS7(k) = sS7v(k);

end

```

% Min

sM7a(k) = 0; sO7a(k) = 0; sW7a(k) = 0; sS7a(k) = 0;

[sO7va(k),sM7va(k),sW7va(k),sS7va(k)] = NASAAsymbrfunc(dVGestM2S(k),20,60,15); %
Min

if sM7va(k) >= Mbr

    sM7a(k) = Mbr;
    Sgn = sign(dVGestM2S(k));
    sO7a(k) = Out*Sgn;
    sW7a(k) = Wght;
    sS7a(k) = FS*Sgn;

elseif sM7va(k) < Mbr

    sM7a(k) = sM7va(k);
    sO7a(k) = sO7va(k);
    sW7a(k) = sW7va(k);
    sS7a(k) = sS7va(k);

end

% Max

sM7b(k) = 0; sO7b(k) = 0; sW7b(k) = 0; sS7b(k) = 0;

[sO7vb(k),sM7vb(k),sW7vb(k),sS7vb(k)] = NASAAsymbrfunc(dVGestP2S(k),20,60,15); %
Max

if sM7vb(k) >= Mbr

    sM7b(k) = Mbr;
    Sgn = sign(dVGestP2S(k));
    sO7b(k) = Out*Sgn;
    sW7b(k) = Wght;
    sS7b(k) = FS*Sgn;

elseif sM7vb(k) < Mbr

    sM7b(k) = sM7vb(k);
    sO7b(k) = sO7vb(k);
    sW7b(k) = sW7vb(k);
    sS7b(k) = sS7vb(k);

end

% Compute nominal Procedure results:

OT(k) = O1(k) + O2(k) + O3(k) + O4(k) + O5(k) + O6(k) + O7(k); % Total output
WT(k) = W1(k) + W2(k) + W3(k) + W4(k) + W5(k) + W6(k) + W7(k); % Total weight
FST(k) = S1(k) + S2(k) + S3(k) + S4(k) + S5(k) + S6(k) + S7(k); % Total firing strength

```

```
if WT(k) == 0
    WT(k) = 1;
end
```

```
NFST(k) = FST(k)/WT(k);           % Normalized total firing strength
NS1(k) = S1(k)/WT(k);
NS2(k) = S2(k)/WT(k);
NS3(k) = S3(k)/WT(k);
NS4(k) = S4(k)/WT(k);
NS5(k) = S5(k)/WT(k);
NS6(k) = S6(k)/WT(k);
NS7(k) = S7(k)/WT(k);
```

% Compute statistical Procedure results:

```
sOT(k) = O1(k) + O2(k) + O3(k) + O4(k) + sO5(k) + O6(k) + sO7(k);   % Total output
sWT(k) = W1(k) + W2(k) + W3(k) + W4(k) + sW5(k) + W6(k) + sW7(k); % Total weight
sFST(k) = S1(k) + S2(k) + S3(k) + S4(k) + sS5(k) + S6(k) + sS7(k);   % Total firing strength
```

```
if sWT(k) == 0
    sWT(k) = 1;
end
```

```
sNFST(k) = sFST(k)/sWT(k);       % Normalized total firing strength
sNS1(k) = S1(k)/sWT(k);
sNS2(k) = S2(k)/sWT(k);
sNS3(k) = S3(k)/sWT(k);
sNS4(k) = S4(k)/sWT(k);
sNS5(k) = sS5(k)/sWT(k);
sNS6(k) = S6(k)/sWT(k);
sNS7(k) = sS7(k)/sWT(k);
```

% Compute min results:

```
sOTa(k) = O1(k) + O2(k) + O3(k) + O4(k) + sO5a(k) + O6(k) + sO7a(k);   % Total
output
sWTa(k) = W1(k) + W2(k) + W3(k) + W4(k) + sW5a(k) + W6(k) + sW7a(k); % Total
weight
sFSTa(k) = S1(k) + S2(k) + S3(k) + S4(k) + sS5a(k) + S6(k) + sS7a(k);   % Total firing
strength
```

```
if sWTa(k) == 0
    sWTa(k) = 1;
end
```

```
sNFSTa(k) = sFSTa(k)/sWTa(k);       % Normalized total firing strength
sNS1a(k) = S1(k)/sWTa(k);
sNS2a(k) = S2(k)/sWTa(k);
sNS3a(k) = S3(k)/sWTa(k);
```

```

sNS4a(k) = S4(k)/sWTa(k);
sNS5a(k) = sS5a(k)/sWTa(k);
sNS6a(k) = S6(k)/sWTa(k);
sNS7a(k) = sS7a(k)/sWTa(k);

% Compute max results:

sOTb(k) = O1(k) + O2(k) + O3(k) + O4(k) + sO5b(k) + O6(k) + sO7b(k);    % Total
output
sWTb(k) = W1(k) + W2(k) + W3(k) + W4(k) + sW5b(k) + W6(k) + sW7b(k);% Total
weight
sFSTb(k) = S1(k) + S2(k) + S3(k) + S4(k) + sS5b(k) + S6(k) + sS7b(k);    % Total firing
strength

if sWTb(k) == 0

    sWTb(k) = 1;

end

sNFSTb(k) = sFSTb(k)/sWTb(k);      % Normalized total firing strength
sNS1b(k) = S1(k)/sWTb(k);
sNS2b(k) = S2(k)/sWTb(k);
sNS3b(k) = S3(k)/sWTb(k);
sNS4b(k) = S4(k)/sWTb(k);
sNS5b(k) = sS5b(k)/sWTb(k);
sNS6b(k) = S6(k)/sWTb(k);
sNS7b(k) = sS7b(k)/sWTb(k);

end

% ----- OUTPUT PLOTS -----

NASASL1plotfunc(ptitle,T,NSD,ds,dh,dVG,dVGest,dVGestM2S,dVGestP2S,mudVG,sigdVG,
...
    M1,M2,M3,M4,M5,M6,M7,OT,O1,O2,O3,O4,O5,O6,O7,WT,W1,W2,W3, ...
    W4,W5,W6,W7,FST,S1,S2,S3,S4,S5,S6,S7,NFST,NS1,NS2,NS3,NS4, ...
    NS5,NS6,NS7,sM5,sM7,sOT,sO5,sO7,sWT,sW5,sW7,sFST,sS5,sS7,sNFST, ...
    sNS1,sNS2,sNS3,sNS4,sNS5,sNS6,sNS7,sM5a,sM7a,sOTa,sO5a,sO7a, ...
    sWTa,sW5a,sW7a,sFSTa,sS5a,sS7a,sNFSTa,sNS1a,sNS2a,sNS3a,sNS4a, ...
    sNS5a,sNS6a,sNS7a,sM5b,sM7b,sOTb,sO5b,sO7b,sWTb,sW5b,sW7b,sFSTb, ...
    sS5b,sS7b,sNFSTb,sNS1b,sNS2b,sNS3b,sNS4b,sNS5b,sNS6b,sNS7b);

Run_time = etime(clock,t0)

```

B.4 fastSL1inpar.m MATLAB Listing

```

% NASAFastSL1inpar.m
% Sets up the input parameters, 2 trajectories, and data bases used by fastSL1in.m
% Assumes that aircraft 1 is nominally ahead of aircraft 2.

```

Seagull Technology, Inc. Proprietary Information

% Developed by: K.Tysen Mueller
% SEAGULL TECHNOLOGY, INC.
% Developed on: 29 September 98
% Modified on: 16 October 98

% ***** INPUTS *****

ptitle = '(JET 1 - JET 2)'; % Common plot title for trajectory pair

%sigr = 0.0625; % Range sigma (nmi) -- Ed Horvath/FAATC
sigr = 0.00823; % Range sigma (nmi) --George Hunter/CTAS Filter Study
sigz = 0.16; % Azimuth sigma (deg) -- both refs.

t01 = 0; % Start time for 1st aircraft trajectory data file
% (must be multiple of delt; sec)
t02 = 0; % Start time for 2nd aircraft trajectory data file
% (must be multiple of delt; sec)

tf1 = 1017.0; % Final time of 1st aircraft trajectory data file
% (must be multiple of delt; sec)
tf2 = 1017.0; % Final time of 2nd aircraft trajectory data file
% (must be multiple of delt; sec)

tiOut = 0; % Start time of output data (must be multiple of delt &
% must not be less than t01 or t02; sec)
tfOut = 540; % Final Time of output data (must be multiple of delt &
% must not be greater than tf1 or tf2; sec)

tbias1 = 54.0; % Additive time biases to 1st and 2nd trajectory time history (sec).
tbias2 = 0; % Value places 1st ahead of 2nd by 3 nm.

sRE1 = 61.02; % Nominal distance for 1st aircraft to Runway Edge from Metering Fix (nm)
sRE2 = 61.02; % Nominal distance for 2nd aircraft to Runway Edge from Metering Fix (nm)

dt = 0.5; % fastS.m trajectory time interval (sec)
delt = 4.5; % Radar sweep interval (sec) -- Note: while the actual
% sweep period has been reported as 4.6 or 4.7 secs,
% the use of an integer multiple value of (dt) is more
% convenient
dtout = 4.5; % Output time interval: should be an integer multiple
% of delt. Note: all the input times listed below have been
% adjusted such that they are integer multiples of dtout.

dsep12 = 3.0; % Min spacing if 1st aircraft is followed by 2nd (nm)
dsep21 = 3.0; % Min spacing if 2nd aircraft is followed by 1st (nm)

xR = 29.0; % TRACON radar east location (nmi) -- origin is SW metering fix.
yR = 23.4; % TRACON radar north location (nmi)
zR = 0; % TRACON radar altitude (kft)

rmax = 17; % Max number of successful radar sweeps to reach range a-b
% filter gain steady-state values
amax = 7; % Max number of successful radar sweeps to reach azimuth a-b

% filter gain steady-state values

% ***** Load TS trajectory *****

% Trajectories have a time interval of dt

load fastTSjet.dat
fastraj1 = fastTSjet; % First trajectory

load fastTSjet.dat
fastraj2 = fastTSjet; % Second trajectory

% Note: For inline trajectories which are based on biased versions of the same
% trajectory, the same nominal trajectory must be loaded twice

% ***** Alpha-Beta Tracking Filter Gains *****

% If there are no false correlations, the gain history is as follows:

FAB =[14 64 64 64 64; 17 54 43 54 43; 22 44 24 44 24; 24 38 16 38 16; ...
26 34 12 36 14; 30 30 9 36 14; 32 27 7 34 12; 34 24 6 34 12; ...
36 22 5 34 12; 40 20 4 34 12; 41 19 3 34 12; 42 18 3 34 12; ...
43 17 3 34 12; 44 16 2 34 12; 45 15 2 34 12; 46 14 2 34 12; ...
47 13 2 34 12];

B.5 fastSL1in.m MATLAB Listing

% **NASAFastSL1in.m**

% Simulation of FAST SL input errors based on the alpha-beta filter errors. It
% takes the ENU time-referenced aircraft trajectory and the radar error statistics
% and computes the time-referenced radar, ENU, time of arrival (ETA), and First
% Common Time Step (FCTS) estimation error mean and sigmas. Based on original
% alphabeta.m and on fastSLin.m simulations. Provides inputs for fastSL1.m.

% developed by: K. Tysen Mueller
% SEAGULL TECHNOLOGY, INC.
% developed on: 29 September 98
% modified on: 6 November 98

%----- INITIALIZATION -----

| | |
|---------------------------|----------------------------|
| D2R = pi/180; | % deg to radians |
| NM2FT = 6076.1033; | % nautical miles to feet |
| HR2S = 3600; | % hours to seconds |
| KTS2FPS = NM2FT/HR2S; | % knots to feet/sec |
| KF2F = 1000; | % kilofeet to feet |
| NM2M = 6076.1033/3.28084; | % Nautical miles to meters |
| M2S = 60; | % Minutes to seconds |
| SXTY42D = 1/64; | % 64'th to decimal |

% Assuming that both trajectories have the same time origin (e.g: time from metering fix):

% Compute adjusted output start time for both trajectories:

t01Out = (tiOut + tbias1);

t02Out = (tiOut + tbias2);

tstart = [t01Out,t02Out];

% Compute adjusted final output time for both trajectories:

tf1Out = (tfOut + tbias1);

tf2Out = (tfOut + tbias2);

tfinal = [tf1Out,tf2Out];

% Compute common adjusted output times:

T0 = min(tstart); % Common initial output time (sec)

TF = min(tfinal); % Common final output time (sec)

ndt1 = ((tf1 - t01)/dtout + 1); % Number of output time points for 1st trajectory

K1 = ((tf1 - t01)/dt + 1); % Number of 1st trajectory points

Ndt1 = ((tf1 - t01)/delt + 1); % Number of 1st trajectory a-b filter gains

ndt2 = ((tf2 - t02)/dtout + 1); % Number of output time points for 2nd trajectory

K2 = ((tf2 - t02)/dt + 1); % Number of 2nd trajectory points

Ndt2 = ((tf2 - t02)/delt + 1); % Number of 2nd trajectory a-b filter gains

k01 = t01Out/dtout; % Common initial time index (1st)

k02 = t02Out/dtout; % Common initial time index (2nd)

K = ((TF - T0)/dtout + 1); % Number of common time steps

rcorrlim = (2*sigr*6076.1033)/(delt^2); % a-b filter absolute value of range acceleration

% correlation limit (ft/s^2) --- computed based on

% maximum range deviation allowed from predicted

% aircraft track, between consecutive radar sweeps

acorrlim = (2*sigz*(pi/180))/(delt^2); % a-b filter absolute value of azimuth acceleration

% correlation limit (deg/s^2) --- computed based on

% maximum azimuth deviation allowed from predicted

% aircraft track, between consecutive radar sweeps

% The basic trajectories are unpacked at dtout rather than at dt. Any time biases are

% applied during the relative aircraft variable calculations

for k = 1:ndt1 % 1st trajectory @ dtout time interval

kk = (k-1)*(dtout/dt) + 1;

t1(k) = (fastraj1(kk,1))/M2S; % Time (min)

x1(k) = fastraj1(kk,2); % East position (ft)

y1(k) = fastraj1(kk,3); % North position (ft)

z1(k) = fastraj1(kk,4); % Altitude (ft)

vx1(k) = fastraj1(kk,5); % East velocity (ft/sec)

vy1(k) = fastraj1(kk,6); % North velocity (ft/sec)

vz1(k) = fastraj1(kk,7); % Altitude rate (ft/sec)

s1(k) = fastraj1(kk,8); % Distance traveled (ft)

psi1(k) = fastraj1(kk,9)*D2R; % Heading, psig (rads)

sdot1(k) = fastraj1(kk,10); % Ground speed, Vg, (ft/sec)

psidot1(k) = fastraj1(kk,11)*D2R; % Heading rate (rads/sec)

sdbldt1(k) = fastraj1(kk,12); % Ground accel (ft/sec^2)

end

```

for k = 1:ndt2                                % 2nd trajectory @ dtout time interval
    kk = (k-1)*(dtout/dt) + 1;
    t2(k) = (fastraj2(kk,1))/M2S;                % Time (min)
    x2(k) = fastraj2(kk,2);                      % East position (ft)
    y2(k) = fastraj2(kk,3);                      % North position (ft)
    z2(k) = fastraj2(kk,4);                      % Altitude (ft)
    vx2(k) = fastraj2(kk,5);                    % East velocity (ft/sec)
    vy2(k) = fastraj2(kk,6);                    % North velocity (ft/sec)
    vz2(k) = fastraj2(kk,7);                    % Altitude rate (ft/sec)
    s2(k) = fastraj2(kk,8);                      % Distance traveled (ft)
    psi2(k) = fastraj2(kk,9)*D2R;                % Heading, psig (rads)
    sdot2(k) = fastraj2(kk,10);                  % Ground speed, Vg, (ft/sec)
    psidot2(k) = fastraj2(kk,11)*D2R;            % Heading rate (rads/sec)
    sdbldt2(k) = fastraj2(kk,12);                % Ground accel (ft/sec^2)
end

% Initialize the alpha-beta filter initial and noise covariance matrices

Cm = zeros(2,2);
Cm(1,1) = (sigr*NM2FT)^2;
Cm(2,2) = (sigz*D2R)^2;

HPh1 = zeros(4,4,ndt1); HPh2 = zeros(4,4,ndt2);
HmuV1 = zeros(4,ndt1); HmuV2 = zeros(4,ndt2);

% ----- COMPUTE TRAJECTORY TRACKING ERROR HISTORIES -----

% ***** 1st Aircraft Trajectory Tracking Errors *****

% Radar estimation error calculations:

[muRN1,muEta1,PrN1] = NASAabfltrfunc(ndt1,delt,x1,y1,z1,vx1,vy1,vz1,psi1, ...
    sdot1,sdbldt1,psidot1,FAB,xR,yR,zR,rcorrlim,acorrlim,rmax,amax,Cm);

% Ground speed, heading, and path estimation error calculations:

[muVG1,muPsiG1,muS1,sigVG1,sigPsiG1,sigS1,HmuV1,HPh1] = ...
    NASAvgfunc(ndt1,dt,dtout,x1,y1,z1,xR,yR,zR,psi1,sdot1,muRN1,muEta1,PrN1);

% ***** 2nd Aircraft Trajectory Tracking Errors *****

% Radar estimation error calculations:

[muRN2,muEta2,PrN2] = NASAabfltrfunc(ndt2,delt,x2,y2,z2,vx2,vy2,vz2,psi2, ...
    sdot2,sdbldt2,psidot2,FAB,xR,yR,zR,rcorrlim,acorrlim,rmax,amax,Cm);

% Ground speed and ETA estimation error calculations:

[muVG2,muPsiG2,muS2,sigVG2,sigPsiG2,sigS2,HmuV2,HPh2] = ...
    NASAvgfunc(ndt2,dt,dtout,x2,y2,z2,xR,yR,zR,psi2,sdot2,muRN2,muEta2,PrN2);

% ----- AIRCRAFT RELATIVE VARIABLE CALCULATIONS -----

```

% The relative aircraft variable calculations are performed by shifting the
 % time index, but not the times, of the respective two trajectories. The
 % effective time of tFCTS is obtained by shifting the nominal tFCTS by tbiasFCTS.

% ***** Aircraft 1 ahead of Aircraft 2 *****

```
T = zeros(1,K);
SD = zeros(1,K); NSD = zeros(1,K);
ds = zeros(1,K); dh = zeros(1,K);
dVG = zeros(1,K); mudVG = zeros(1,K); sigdVG = zeros(1,K);
```

% Compute relative variables:

for k = 1:K

```
    k1 = k01 + k;
    k2 = k02 + k;
    T(k) = (T0 + (k-1)*dtout)/M2S;
```

% Relative ground speed (kts) statistics:

```
    dVG(k) = (sdot1(k1) - sdot2(k2))/KTS2FPS;      % Nominal
    mudVG(k) = muVG1(k1) - muVG2(k2);              % Mean
    sigdVG(k) = sqrt(sigVG1(k1)^2 + sigVG2(k2)^2); % Sigma
    dVGest(k) = (dVG(k) + mudVG(k));                % Estimate
    dVGestM2S(k) = dVGest(k) - 2*sigdVG(k);          % Estimate - 2*sigma
    dVGestP2S(k) = dVGest(k) + 2*sigdVG(k);          % Estimate + 2*sigma
```

% Nominal relative variables:

```
    S1(k) = (sRE1 - (s1(k1)/NM2FT));      % Path distances wrt runway edge (nm)
    S2(k) = (sRE2 - (s2(k2)/NM2FT));
    SD(k) = (S2(k) - S1(k));               % Relative (path) separation (nm)

    if SD(k) > 0
        NSD(k) = SD(k)/dsep12;             % NSD Track wrt RE
    else
        NSD(k) = SD(k)/dsep21;
    end

    dh(k) = z2(k2) - z1(k1);               % Relative altitude (ft)
    ds(k) = (sqrt((x1(k1) - x2(k2))^2 + ...
        (y1(k1) - y2(k2))^2))/NM2FT;       % Relative (line-of-sight) separation (nm)
end
```

% Output data:

fastSL1stats = zeros(K,7);

```
    fastSL1stats(:,1) = T(:);               % Time (min)
    fastSL1stats(:,2) = NSD(:);              % Nominal NSD Track
    fastSL1stats(:,3) = ds(:);               % Nominal relative separation (nm)
```

```
fastSL1stats(:,4) = dh(:);           % Nominal relative altitude (ft)
fastSL1stats(:,5) = dVG(:);          % Nominal relative ground speed (kts)
fastSL1stats(:,6) = mudVG(:);        % Relative ground speed mean (kts)
fastSL1stats(:,7) = sigdVG(:);       % Relative ground speed sigma (kts)
```

% ----- OUTPUT FILE -----

```
save fastSL1stats.datfastSL1stats -ascii      % Aircraft 1 ahead of 2 stats
```

B.6 abfltrfunc.m MATLAB Listing

% **NASAabfltrfunc.m**

% Computes the radar tracking error statistics for a nominal fastTS.m trajectory

% Input: All inputs @ delt

```
% ndt = number of trajectory time points
% delt = radar sweep interval -- integer number of dt (sec)
% x,y,z = east, north, vertical position of aircraft (ft)
% xR,yR,zR = east, north, vertical position of radar (nm)
% vx,vy,vz = east, north, vertical velocity of aircraft (ft/sec)
% psi,psidot = heading and heading rate (deg, deg/sec)
% sdot,sdbldt = ground speed and acceleration (ft/sec, ft/sec^2)
% Cm = radar measurement noise (range (ft) & azimuth (rads)) 2x2 matrix
% rcorrlim,acorrlim = a-b filter absolute value of range and acceleration
%                  correlation limits -- if exceeded, unsuccessful
%                  correlation results for this track (ft/s^2, deg/s^2)
% rmax,amax = max number of successful radar sweeps to reach range a-b
%            filter gain steady-state range and azimuth values
% FAB = contains a-b filter gains (units of 1/64 th) as function of firmness:
%     FAB(:,1) = firmness (strength of correlation of current with past tracks)
%     FAB(:,2) = range gain, FAB(:,3) = range rate gain
%     FAB(:,4) = azimuth gain, FAB(:,5) = azimuth rate gain
```

% Output: All outputs @ delt

```
% muRN = mean radar estimation error history matrix
%     muRN(1,:) = mean range (ft), muRN(2,:) = range rate (ft/sec),
%     muRN(3,:) = azimuth (rads), muRN(4,:) = azimuth rate (rads/sec)
% muEta = mean altitude (ft) & altitude rate (ft/sec) estimation error history
%     muEta(eta(:),etadot(:),:)
% PrN = radar estimation error covariance history matrix
%     PrN(1,1,:) = range variance (ft^2),
%     PrN(2,2,:) = range rate variance (ft/sec)^2,
%     PrN(3,3,:) = azimuth variance (rads^2),
%     PrN(4,4,:) = azimuth rate variance (rads/sec)^2
```

```
function[muRN,muEta,PrN] = ...
    NASAabfltrfunc(ndt,delt,x,y,z,vx,vy,vz,psi,sdot,sdbldt,psidot,FAB, ...
    xR,yR,zR,rcorrlim,acorrlim,rmax,amax,Cm)
```

```
D2R = pi/180;                                % deg to radians
```

```

NM2FT = 6076.1033;           % nautical miles to feet
HR2S = 3600;                 % hours to seconds
KTS2FPS = NM2FT/HR2S;        % knots to feet/sec
SXTY42D = 1/64;              % 64'th to decimal

cntR = rmax;
cntAz = amax;

for i = 1:ndt
% Compute ENU radar coordinate trajectory
    xr = x(i) - xR*NM2FT;
    yr = y(i) - yR*NM2FT;
    zr = z(i) - zR*NM2FT;
    vxr = vx(i);
    vyr = vy(i);
    vzr = vz(i);
    r = sqrt(xr^2 + yr^2 + zr^2);
    theta = atan2(xr,yr);
    rdot = sdot(i)*cos(psi(i) - theta);

% Compute modeling errors:
    if r ~= 0                 % Check to make sure that range is non-zero
        thetadt = (sdot(i)/r)*sin(psi(i) - theta);
    % Exact solutions:
        eta(i) = (sqrt(1 - (zr/r)^2) - 1)*r;
        etadot(i) = ((1/sqrt(1 - (zr/r)^2)) - 1)*rdot - ...
            ((zr/r)/sqrt(1 - (zr/r)^2))*vzr;

        ar(i) = sdbldt(i)*cos(psi(i) - theta) - sdot(i)*psidot(i)* ...
            sin(psi(i) - theta) + ((sdot(i)^2)/r)*(sin(psi(i) - theta))^2;
        atheta(i) = (sdbldt(i)*sin(psi(i) - theta) + sdot(i)*psidot(i)* ...
            cos(psi(i) - theta) - ((sdot(i)^2)/r)*(sin(2*(psi(i) - theta))))/r;
    else
        thetadt = 0;
        eta(i) = 0;
        etadot(i) = 0;
        ar(i) = sdbldt(i)*cos(psi(i) - theta) - sdot(i)*psidot(i)* ...
            sin(psi(i) - theta);
        atheta(i) = 0;
    end

% muEta(:,i) = [eta(i); etadot(i)]; % Error = (Nominal - Estimate) convention
% muAcc = [ar(i); atheta(i)];
muEta(:,i) = -[eta(i); etadot(i)]; % Error = (Estimate - Nominal) convention
muAcc = -[ar(i); atheta(i)];

R(i) = r/NM2FT;
Rdot(i) = rdot/KTS2FPS;
Theta(i) = theta/D2R;
Thetadot(i) = thetadt/D2R;
Atheta(i) = r*atheta(i);

% Select alpha-beta gain based on whether a radial or azimuth acceleration is
% present -- assumes that accelerations lead to loss of track correlations:

```

```

if abs(ar(i)) > rcorrlim
    cntR = cntR - 2;
else
    cntR = cntR + 1;
end

if cntR > rmax
    cntR = rmax;
elseif cntR < 1
    cntR = 1;
end

if abs(atheta(i)) > acorrlim
    cntAz = cntAz - 1;
else
    cntAz = cntAz + 1;
end

if cntAz > amax
    cntAz = amax;
elseif cntAz < 1
    cntAz = 1;
end

alphan(i) = FAB(cntR,2)*SXTY42D; % Convert to decimal
betar(i) = FAB(cntR,3)*SXTY42D;
alphaz(i) = FAB(cntAz,4)*SXTY42D;
betaz(i) = FAB(cntAz,5)*SXTY42D;

```

% Alpha-Beta Filter calculations.

% Note: While the horizontal covariance matrix and corresponding ETA
 % variance history will be computed every dtout seconds, the alpha-beta
 % filter runs at the ARTS radar sweep period, delt. The actual period is
 % 4.6 secs. For convenience, it is assumed that it is 4.5 secs.

```

A(1,1) = 1 - alphan(i);
A(1,2) = (1 - alphan(i))*delt;
A(2,1) = -(betar(i)/delt);
A(2,2) = 1 - betar(i);
A(3,3) = 1 - alphaz(i);
A(3,4) = (1 - alphaz(i))*delt;
A(4,3) = -(betaz(i)/delt);
A(4,4) = 1 - betaz(i);

B(1,1) = (1 - alphan(i))*((delt)^2)/2;
B(2,1) = (1 - (betar(i)/2))*delt;
B(3,2) = (1 - alphaz(i))*((delt)^2)/2;
B(4,2) = (1 - (betaz(i)/2))*delt;

C(1,1) = -alphan(i);
C(2,1) = -(betar(i)/delt);
C(3,2) = -alphaz(i);
C(4,2) = -(betaz(i)/delt);

```

```

if i ==1
    muRN(:,i) = B*muAcc;
    PrN(:,i) = (C*Cm*(C'));
else
    muRN(:,i) = A*muRO + B*muAcc;
    PrN(:,i) = (A*PrO*(A') + C*Cm*(C'));
end

muR(i) = muRN(1)/NM2FT;
muRDT(i) = muRN(2)/KTS2FPS;
muA(i) = muRN(3)/D2R;
muADT(i) = muRN(4)/D2R;
muRO = muRN(:,i);

sigR(i) = (sqrt(PrN(1,1)))/NM2FT;
sigRDT(i) = (sqrt(PrN(2,2)))/KTS2FPS;
sigA(i) = (sqrt(PrN(3,3)))/D2R;
sigADT(i) = (sqrt(PrN(4,4)))/D2R;

PrO = PrN(:,i);
end

```

B.7 vgfunc.m MATLAB Listing

```

% NASAvgfunc.m
% Computes ground speed, heading, and path statistics

% Input: All inputs in @ dtout except INT which is @ dt
% ndt = max number of trajectory points @ dtout
% dt = fastTS.m trajectory & INT time interval (sec)
% dtout = input trajectory time interval -- multiple of dt (sec)
% x,y,z = east, north, vertical nominal position history of aircraft (ft)
% xR,yR,zR = east, north, vertical position of radar (nm)
% psi = heading (deg)
% sdot = ground speed (ft/sec)
% muRN = mean radar estimation error history matrix
%     muRN(1,:) = mean range (ft), muRN(2,:) = range rate (ft/sec),
%     muRN(3,:) = azimuth (rads), muRN(4,:) = azimuth rate (rads/sec)
% muEta = mean altitude (ft) & altitude rate (ft/sec) estimation error history
%     muEta(eta(:),etadot(:),:)
% PrN = radar estimation error covariance history matrix
%     PrN(1,1,:) = range variance (ft^2),
%     PrN(2,2,:) = range rate variance (ft/sec)^2,
%     PrN(3,3,:) = azimuth variance (rads^2),
%     PrN(4,4,:) = azimuth rate variance (rads/sec)^2

% Output: All output @ dtout
% muVG = mean ground speed history (kts)
% muPsiG = mean heading history (deg)
% muS = mean path history (nm)
% sigVG = ground speed sigma history (kts)

```

```
% sigPsiG = heading sigma history (deg)
% sigS = path sigma history (nm)
% HmuV = path and ground speed mean history (ft,ft/sec)
% HPh = path and ground speed covariance history matrix (ft,ft/sec)
```

```
function[muVG,muPsiG,muS,sigVG,sigPsiG,sigS,HmuV,HPh] = ...
    NASAvgfunc(ndt,dt,dtout,x,y,z,xR,yR,zR,psi,sdot,muRN,muEta,PrN)
```

```
D2R = pi/180; % deg to radians
NM2FT = 6076.1033; % nautical miles to feet
HR2S = 3600; % hours to seconds
KTS2FPS = NM2FT/HR2S; % knots to feet/sec
KF2F = 1000; % kilofeet to feet
NM2M = 6076.1033/3.28084; % Nautical miles to meters
M2S = 60; % Minutes to seconds
SXTY42D = 1/64; % 64'th to decimal
```

```
for i = 1:ndt
```

```
% Compute Hr , Heta, Hs, and S matrices
```

```
Hr = zeros(4,4);
Heta = zeros(4,2);
Hs = zeros(2,4);
H = zeros(2,4);
```

```
xr = x(i) - xR*NM2FT ;
yr = y(i) - yR*NM2FT ;
zr = z(i) - zR*NM2FT ;
r = sqrt(xr^2 + yr^2 + zr^2);
theta = atan2(xr,yr);
rdot = sdot(i)*cos(psi(i) - theta);
```

```
if r ~= 0 % Check to make sure that range is non-zero
    thetadt = (sdot(i)/r)*sin(psi(i) - theta);
end
```

```
Hr(1,1) = sin(theta);
Hr(1,3) = r*cos(theta);
Hr(2,1) = cos(theta);
Hr(2,3) = -r*sin(theta);
Hr(3,1) = thetadt*cos(theta);
Hr(3,2) = sin(theta);
Hr(3,3) = (rdot*cos(theta) - r*thetadt*sin(theta));
Hr(3,4) = r*cos(theta);
Hr(4,1) = -thetadt*sin(theta);
Hr(4,2) = cos(theta);
Hr(4,3) = -(rdot*sin(theta) + r*thetadt*cos(theta));
Hr(4,4) = -r*sin(theta);
```

```
Heta(1,1) = sin(theta);
Heta(2,1) = cos(theta);
```

```
Heta(3,1) = thetadt*cos(theta);
Heta(3,2) = sin(theta);
Heta(4,1) = -thetadt*sin(theta);
Heta(4,2) = cos(theta);
```

```
Hs(1,1) = sin(psi(i));
Hs(1,2) = cos(psi(i));
Hs(2,3) = sin(psi(i));
Hs(2,4) = cos(psi(i));
```

```
H(1,3) = sin(psi(i));
H(1,4) = cos(psi(i));
H(2,3) = cos(psi(i))/sdot(i);
H(2,4) = -sin(psi(i))/sdot(i);
```

% Compute path, ground speed, and heading statistics

```
muV = Hr*muRN(:,i) + Heta*muEta(:,i); % Horizontal position & velocity mean
muVg = H*muV; % Ground speed & heading mean
muP = Hs*muV; % Path and ground speed mean
```

```
muS(i) = muP(1)/NM2FT; % Mean path error (nm)
muVG(i) = muVg(1)/KTS2FPS; % Mean ground speed error (kts)
muPsiG(i) = muVg(2)/D2R; % Mean heading error (deg)
```

```
Ph = Hr*PrN(:,i)*(Hr'); % Horizontal position & velocity covariance matrix
PVg = H*Ph*(H'); % Ground speed & heading covariance matrix
PS = Hs*Ph*(Hs'); % Path and ground speed covariance matrix
```

```
HPh(:,i) = PS(:,i); % Ground path and speed statistics for FCTS calculations
HmuV(:,i) = muP(:,i);
```

```
sigS(i) = (sqrt(PS(1,1)))/NM2FT; % Path sigma (nm)
sigVG(i) = sqrt(PVg(1,1))/KTS2FPS; % Ground speed sigma (kts)
sigPsiG(i) = sqrt(PVg(2,2))/D2R; % Heading sigma (deg)
```

end

B.8 symbrfunc.m MATLAB Listing

% **NASAsymbrfunc.m**

% Computes the membership value, output value, weight, and the product
% of output and weight (the firing strength) for a symmetric (about
% vertical axis) trapezoidal membership function.

% Assumption: 1) Symmetric trapezoidal membership function
% 2) a and b are positive with a < b
% 3) membership is zero for x between -a and +a
% 4) membership is unity for x < -b and for x > +b

% Input: x = input value
% a = membership function lower limit (membership = 0)

% b = membership function upper limit (membership = 1)
 % c = consequent function output value

% Output: Out = proposition output value
 % Mbr = membership value,
 % Wght = weight corresponding to membership value
 % FS = firing strength (weighted output)

function [Out,Mbr,Wght,FS] = NASAsymbrfunc(x,a,b,c)

Out = 0;
 Mbr = 0;
 Wght = 0;
 FS = 0;

if a ~= b % Fuzzy Logic

 if x >= a

 Out = c;

 if x < b

 Mbr = (x - a)/(b - a);
 Wght = -5*(x - a)*(x + a - 2*b)/(b - a)^2;
 FS = c*Wght;

 elseif x >= b

 Mbr = 1;
 Wght = 5;
 FS = 5*c;

 end

elseif x < -a

 Out = -c;

 if x > -b

 Mbr = -(x + a)/(b - a);
 Wght = -5*(x + a)*(x - a + 2*b)/(b - a)^2;
 FS = -c*Wght;

 elseif x <= -b

 Mbr = 1;
 Wght = 5;
 FS = -5*c;

 end

end

end

B.9 vgscrfunc.m MATLAB Listing

```
% NASAvgscrfunc.m
% Computes mean output, membership, weight, and firing strength for
% relative ground statistics. Proposition constants correspond to
% 'Is Faster' membership function and 'Is Marginally Favored'
% consequent function.

% Assumption: 1) Symmetric trapezoidal membership function

% Input:  VGRest = estimate of relative ground speed (kts)
%         sigVGR = relative ground speed sigma (kts)
%         a,b = start and end of trapezoidal ramp
%         c = output value

% Output: VGOUT = consequent function mean output value
%         MBR = mean membership value
%         WGHT = consequent function mean weight
%         VGSCORE = mean firing strength

function[VGOUT,MBR,WGHT,VGSCORE] = NASAvgscrfunc(VGRest,sigVGR,a,b,c)

VGOUT = 0;
MBR = 0;
WGHT = 0;
VGSCORE = 0;
pVGa = 0;
pVGb = 0;
SQRT2 = sqrt(2);

% Check how narrow probability density function is:

    if sigVGR > 0.4          % (kts)

% Decision Error Probabilities:
    pVGa = (0.5)*(erfc((a + VGRest)/(SQRT2*sigVGR)));
    pVGb = (0.5)*(erfc((a - VGRest)/(SQRT2*sigVGR)));

% Expected Output Values:
    VGaout = -c*pVGa;          % Case a output value
    VGbout = c*pVGb;          % Case b output value
    VGOUT = VGbout + VGaout;   % Case total output value

% Expected Membership and Weight:
    v = 0;
    dv = (b - a)/200;          % integration step size (kts)

    VGaMBR = 0;
    VGbMBR = 0;
```

```

VGaWGHT = 0;
VGbWGHT = 0;

cMBR = 1/((b - a)*sigVGR*sqrt(2*pi)); % membership integral scale factor
cWGHT = -5/(((b - a)^2)*sigVGR*sqrt(2*pi)); % weight integral scale factor

dnm = (2*sigVGR^2); % exponent denominator

% Weight and membership integrals calculation: combines both
% <-a and >+a calculations based on a change of variables
% in the former integral:

for n=a:dv:(b-dv)

    v = n;
    zMBR = (v - a);
    zWGHT = (v - a)*(v + a - 2*b);

    ua = ((v + VGRest)^2)/dnm;
    Expnta = exp(-ua);
    VGaMBR = VGaMBR + dv*cMBR*zMBR*Expnta;
    VGaWGHT = VGaWGHT + dv*cWGHT*zWGHT*Expnta;

    ub = ((v - VGRest)^2)/dnm;
    Expntb = exp(-ub);
    VGbMBR = VGbMBR + dv*cMBR*zMBR*Expntb;
    VGbWGHT = VGbWGHT + dv*cWGHT*zWGHT*Expntb;

end

ExpntaF = 0.5*erfc((b + VGRest)/(SQRT2*sigVGR));

VGaMBR = VGaMBR + ExpntaF; % Case a mean membership
VGaWGHT = VGaWGHT + 5*ExpntaF; % Case a mean weight

ExpntbF = 0.5*erfc((b - VGRest)/(SQRT2*sigVGR));

VGbMBR = VGbMBR + ExpntbF; % Case b mean membership
VGbWGHT = VGbWGHT + 5*ExpntbF; % Case b mean weight

MBR = VGaMBR + VGbMBR; % mean membership
WGHT = VGaWGHT + VGbWGHT; % mean weight

% Expected Firing Strength:

VGaSCORE = -c*VGaWGHT;
VGbSCORE = c*VGbWGHT;
VGSCORE = VGaSCORE + VGbSCORE; % mean firing strength

else

% If sigVGR is too small relative to VGRest, the expected value
% integrals, above, behave like a Dirac delta function whose output is

```

% only determined by the estimate, VGRest. If no numerical integration
% were required, then this situation would be handled automatically
% by the above logic

```

    if VGRest >= a

        VGOUT = c;

        if VGRest <= b

            MBR = (VGRest - a)/(b - a);
            WGHT = -5*(VGRest - a)*(VGRest + a - 2*b)/((b - a)^2);

        else

            MBR = 1;
            WGHT = 5;

        end

    elseif VGRest <= -a

        VGOUT = -c;

        if VGRest >= -b

            MBR = -(VGRest + a)/(b - a);
            WGHT = -5*(VGRest + a)*(VGRest - a + 2*b)/((b - a)^2);

        else

            MBR = 1;
            WGHT = 5;

        end

    end

    VGSCORE = VGOUT*WGHT;

end

```

B.10 mbrfunc.m MATLAB Listing

```

% NASAmbrfunc.m
% Computes the membership value, output value, weight, and the product
% of the output and weight (the firing strength) for a one-sided
% trapezoidal membership function.
% Assumptions: 1) Either a and b are both positive or both negative
%               2) a is nearest zero while b is furthest from zero
%               3) a ~ b
%               4) The value of c associated with a and b is specified

```

```
%          by the input -- no sign reversals are made for c if
%          a and b are negative.
```

```
% Input:  x = input value
%          a = membership function input lower limit
%          am = membership function value at lower input limit
%          b = membership function input upper limit
%          bm = membership function value at upper input limit
%          c = consequent function output value.
```

```
% Output: Out = proposition output,
%          Mbr = membership value
%          Wght = weight
%          FS = firing strength.
```

```
function [Out,Mbr,Wght,FS] = NASAmbrfunc(x,a,am,b,bm,c)
```

```
Out = 0;
Mbr = 0;
Wght = 0;
FS = 0;
```

```
if a ~= b          % Fuzzy Logic
```

```
    if (b > 0)      % a and b are positive
```

```
        if (x > a) & (x < b)
```

```
            Out = c;
```

```
            if (am == 0) & (bm == 1)
```

```
                Mbr = (x - a)/(b - a);
                Wght = -5*(x - a)*(x + a - 2*b)/(b - a)^2;
                FS = c*Wght;
```

```
            elseif (am == 1) & (bm == 0)
```

```
                Mbr = 1 - (x - a)/(b - a);
                Wght = -5*(x - b)*(x - 2*a + b)/(b - a)^2;
                FS = c*Wght;
```

```
            end
```

```
        elseif x <= a
```

```
            Mbr = am;
            Wght = 5*am;
            FS = 5*c*am;
            Out = c*am;
```

```
        elseif x >= b
```

```
            Mbr = bm;
```

```

        Wght= 5*bm;
        FS = 5*c*bm;
        Out = c*bm;

    end

elseif (b < 0)      % a and b are negative

    if (x < a) & (x > b)

        Out = c;

        if (am == 0) & (bm == 1)

            Mbr = (x - a)/(b - a);
            Wght = -5*(x - a)*(x + a - 2*b)/(b - a)^2;
            FS = c*Wght;

        elseif (am == 1) & (bm == 0)

            Mbr = 1 - (x - a)/(b - a);
            Wght = -5*(x - b)*(x - 2*a + b)/(b - a)^2;
            FS = c*Wght;

        end

    elseif x >= a

        Mbr = am;
        Wght = 5*am;
        FS = 5*c*am;
        Out = c*am;

    elseif x <= b

        Mbr = bm;
        Wght = 5*bm;
        FS = 5*c*bm;
        Out = c*bm;

    end
end
end

```

B.11 SL1plotfunc.m MATLAB Listing

```

% NASASL1plotfunc.m
% Plots relative statistics computed by fastSL1.m for two aircraft

% Created by:      K. Tysen Mueller
%                  SEAGULL TECHNOLOGY, INC.
% Created on:      1 October 98

```

% Modified on: 27 October 98

```
function[] = NASASL1plotfunc(ptitle,T,NSD,ds,dh,dVG,dVGest,dVGestM2S,dVGestP2S, ...
    mudVG,sigdVG,M1,M2,M3,M4,M5,M6,M7,OT,O1,O2,O3,O4,O5,O6,O7, ...
    WT,W1,W2,W3,W4,W5,W6,W7,FST,S1,S2,S3,S4,S5,S6,S7,NFST,NS1, ...
    NS2,NS3,NS4,NS5,NS6,NS7,sM5,sM7,sOT,sO5,sO7,sWT,sW5,sW7, ...
    sFST,sS5,sS7,sNFST,sNS1,sNS2,sNS3,sNS4,sNS5,sNS6,sNS7, ...
    sM5a,sM7a,sOTa,sO5a,sO7a,sWTa,sW5a,sW7a,sFSTa,sS5a,sS7a, ...
    sNFSTa,sNS1a,sNS2a,sNS3a,sNS4a,sNS5a,sNS6a,sNS7a,sM5b,sM7b, ...
    sOTb,sO5b,sO7b,sWTb,sW5b,sW7b,sFSTb,sS5b,sS7b,sNFSTb,sNS1b, ...
    sNS2b,sNS3b,sNS4b,sNS5b,sNS6b,sNS7b);
```

```
subplot(4,1,1)
plot(T(:), NSD(:),'b')
title(['NOMINAL VARIABLES ',ptitle])
ylabel('NSD TRACK')
axis([0 9 0 2])
grid on
```

```
subplot(4,1,2)
plot(T(:), ds(:),'b')
ylabel('REL SEP (nm)')
axis([0 9 0 5])
grid on
```

```
subplot(4,1,3)
plot(T(:), dh(:),'b')
ylabel('REL ALT (ft)')
axis([0 9 -1000 3000])
grid on
```

```
subplot(4,1,4)
plot(T(:), dVG(:),'b')
ylabel('REL G SPEED (kts)')
xlabel('TIME FROM METERING FIX (min)')
axis([0 9 -50 10])
grid on
```

figure

```
subplot(4,1,1)
plot(T(:), dVG(:),'b')
title(['REL GROUND SPEED STATISTICS (knots) ',ptitle])
ylabel('NOMINAL')
axis([0 9 -80 40])
grid on
```

```
subplot(4,1,2)
plot(T(:), dVGest(:),'b',T(:), dVGestM2S(:),'--m', ...
    T(:), dVGestP2S(:),'--m');
ylabel('EST +/- 2*SIG')
axis([0 9 -80 40])
grid on
```

```
subplot(4,1,3)
plot(T(:), mudVG(:),'b')
ylabel('ERROR MEAN')
axis([0 9 -80 40])
grid on
```

```
subplot(4,1,4)
plot(T(:), sigdVG(:),'b')
ylabel('ERROR SIGMA')
xlabel('TIME FROM METERING FIX (min)')
axis([0 9 0 20])
grid on
```

figure

```
subplot(4,1,1)
plot(T(:), OT(:),T(:), sOT(:),'m--',T(:), sOTa(:),'k:',T(:), sOTb(:),'k:')
title(['ORDERING PROCEDURE RESULTS ',ptitle])
ylabel('OUTPUT')
axis([0 9 0 100])
grid on
```

```
subplot(4,1,2)
plot(T(:), WT(:),T(:), sWT(:),'m--',T(:), sWTa(:),'k:',T(:), sWTb(:),'k:')
ylabel('WEIGHT')
axis([0 9 0 30])
grid on
```

```
subplot(4,1,3)
plot(T(:), FST(:),T(:), sFST(:),'m--',T(:), sFSTa(:),'k:',T(:), sFSTb(:),'k:')
ylabel('FIRE STRENGTH')
axis([0 9 0 400])
grid on
```

```
subplot(4,1,4)
plot(T(:), NFST(:),T(:), sNFST(:),'m--',T(:), sNFSTa(:),'k:',T(:), sNFSTb(:),'k:')
ylabel('N. FIRE STRENGTH')
xlabel('TIME FROM METERING FIX (min)')
axis([0 9 0 20])
grid on
```

figure

```
subplot(4,1,1)
plot(T(:), O5(:),T(:), sO5(:),'m--',T(:), sO5a(:),'k:',T(:), sO5b(:),'k:')
title(['"IS FASTER" PROPOSITION RESULTS ', ptitle])
ylabel('OUTPUT')
axis([0 9 -20 20])
grid on
```

```
subplot(4,1,2)
plot(T(:), W5(:),T(:), sW5(:),'m--',T(:), sW5a(:),'k:',T(:), sW5b(:),'k:')

```

```
ylabel('WEIGHT')
axis([0 9 -2 6])
grid on
```

```
subplot(4,1,3)
plot(T(:), S5(:),T(:), sS5(:),'m--',T(:), sS5a(:),'k:',T(:), sS5b(:),'k:')
ylabel('FIRE STRNGTH')
axis([0 9 -50 50])
grid on
```

```
subplot(4,1,4)
plot(T(:), NS5(:),T(:), sNS5(:),'m--',T(:), sNS5a(:),'k:',T(:), sNS5b(:),'k:')
ylabel('N. FIRE STRNGTH')
xlabel('TIME FROM METERING FIX (min)')
axis([0 9 -5 10])
grid on
```

figure

```
subplot(4,1,1)
plot(T(:), O7(:),T(:), sO7(:),'m--',T(:), sO7a(:),'k:',T(:), sO7b(:),'k:')
title(['"IS CLOSE & FASTER @ FCTS" PROPOSITION RESULTS ',ptitle])
ylabel('OUTPUT')
axis([0 9 -20 20])
grid on
```

```
subplot(4,1,2)
plot(T(:), W7(:),T(:), sW7(:),'m--',T(:), sW7a(:),'k:',T(:), sW7b(:),'k:')
ylabel('WEIGHT')
axis([0 9 -1 5])
grid on
```

```
subplot(4,1,3)
plot(T(:), S7(:),T(:), sS7(:),'m--',T(:), sS7a(:),'k:',T(:), sS7b(:),'k:')
ylabel('FIRE STRNGTH')
axis([0 9 -50 50])
grid on
```

```
subplot(4,1,4)
plot(T(:), NS7(:),T(:), sNS7(:),'m--',T(:), sNS7a(:),'k:',T(:), sNS7b(:),'k:')
ylabel('N. FIRE STRNGTH')
axis([0 9 -5 5])
xlabel('TIME FROM METERING FIX (min)')
grid on
```

figure

```
subplot(7,1,1)
plot(T(:), S1(:))
title(['ORDERING PROCEDURE FIRING STRENGTH HISTORIES ',ptitle])
ylabel('FS 1')
axis([0 9 -50 150])
grid on
```

```
subplot(7,1,2)
plot(T(:), S2(:))
ylabel('FS 2')
axis([0 9 -50 150])
grid on
```

```
subplot(7,1,3)
plot(T(:), S3(:))
ylabel('FS 3')
axis([0 9 -50 150])
grid on
```

```
subplot(7,1,4)
plot(T(:), S4(:))
ylabel('FS 4')
axis([0 9 -50 150])
grid on
```

```
subplot(7,1,5)
plot(T(:), S5(:),T(:), sS5(:),'m--')
ylabel('FS 5')
axis([0 9 -50 150])
grid on
```

```
subplot(7,1,6)
plot(T(:), S6(:))
ylabel('FS 6')
axis([0 9 -50 150])
grid on
```

```
subplot(7,1,7)
plot(T(:), S7(:),T(:), sS7(:),'m--')
ylabel('FS 7')
xlabel('TIME FROM METERING FIX (min)')
axis([0 9 -50 150])
grid on
```

figure

```
subplot(7,1,1)
plot(T(:), NS1(:),T(:), sNS1(:), 'm--')
title(['ORDERING PROCEDURE NORMALIZED FIRING STRENGTH HISTORIES ', ptitle])
ylabel('NFS 1')
axis([0 9 -5 10])
grid on
```

```
subplot(7,1,2)
plot(T(:), NS2(:),T(:), sNS2(:), 'm--')
ylabel('NFS 2')
axis([0 9 -5 10])
grid on
```

```
subplot(7,1,3)
plot(T(:), NS3(:),T(:), sNS3(:), 'm--')
```

```

ylabel('NFS 3')
axis([0 9 -5 10])
grid on

subplot(7,1,4)
plot(T(:),NS4(:),T(:), sNS4(:), 'm--')
ylabel('NFS 4')
axis([0 9 -5 10])
grid on

subplot(7,1,5)
plot(T(:), NS5(:),T(:), sNS5(:), 'm--')
ylabel('NFS 5')
axis([0 9 -5 10])
grid on

subplot(7,1,6)
plot(T(:), NS6(:),T(:), sNS6(:), 'm--')
ylabel('NFS 6')
axis([0 9 -5 10])
grid on

subplot(7,1,7)
plot(T(:), NS7(:),T(:), sNS7(:), 'm--')
ylabel('NFS 7')
xlabel('TIME FROM METERING FIX (min)')
axis([0 9 -5 10])
grid on

figure

subplot(4,1,1)
plot(T(:), NSD(:))
title(['ORDERING PROCEDURE MEMBERSHIP INPUT HISTORIES ', ptitle])
ylabel('NSD')
axis([0 9 0 2])
grid on

subplot(4,1,2)
plot(T(:), ds(:))
ylabel('SD (nmi)')
axis([0 9 0 5])
grid on

subplot(4,1,3)
plot(T(:), dh(:))
ylabel('HD (ft)')
axis([0 9 -1000 3000])
grid on

subplot(4,1,4)
plot(T(:), dVG(:),T(:), dVGest(:),'m--')
ylabel('VGD (kts)')
xlabel('TIME FROM METERING FIX (min)')

```

```
axis([0 9 -80 40])
grid on
```

figure

```
subplot(7,1,1)
plot(T(:), M1(:))
title(['ORDERING PROCEDURE MEMBERSHIP HISTORIES ', ptitle])
ylabel('M 1')
axis([0 9 -0.2 1.2])
grid on
```

```
subplot(7,1,2)
plot(T(:), M2(:))
ylabel('M 2')
axis([0 9 -0.2 1.2])
grid on
```

```
subplot(7,1,3)
plot(T(:), M3(:))
ylabel('M 3')
axis([0 9 -0.2 1.2])
grid on
```

```
subplot(7,1,4)
plot(T(:), M4(:))
ylabel('M 4')
axis([0 9 -0.2 1.2])
grid on
```

```
subplot(7,1,5)
plot(T(:), M5(:), T(:), sM5(:), 'm--')
ylabel('M 5')
axis([0 9 -0.2 1.2])
grid on
```

```
subplot(7,1,6)
plot(T(:), M6(:))
ylabel('M 6')
axis([0 9 -0.2 1.2])
grid on
```

```
subplot(7,1,7)
plot(T(:), M7(:), T(:), sM7(:), 'm--')
ylabel('M 7')
xlabel('TIME FROM METERING FIX (min)')
axis([0 9 -0.2 1.2])
grid on
```

figure

```
subplot(7,1,1)
plot(T(:), O1(:))
title(['ORDERING PROCEDURE OUTPUT HISTORIES ', ptitle])
```

```

ylabel('O 1')
axis([0 9 -20 40])
grid on

subplot(7,1,2)
plot(T(:), O2(:))
ylabel('O 2')
axis([0 9 -20 40])
grid on

subplot(7,1,3)
plot(T(:), O3(:))
ylabel('O 3')
axis([0 9 -20 40])
grid on

subplot(7,1,4)
plot(T(:), O4(:))
ylabel('O 4')
axis([0 9 -20 40])
grid on

subplot(7,1,5)
plot(T(:), O5(:),T(:), sO5(:),'m--')
ylabel('O 5')
axis([0 9 -20 40])
grid on

subplot(7,1,6)
plot(T(:), O6(:))
ylabel('O 6')
axis([0 9 -20 40])
grid on

subplot(7,1,7)
plot(T(:), O7(:),T(:), sO7(:),'m--')
ylabel('O 7')
xlabel('TIME FROM METERING FIX (min)')
axis([0 9 -20 40])
grid on

figure

subplot(7,1,1)
plot(T(:), W1(:))
title(['ORDERING PROCEDURE WEIGHT HISTORIES ',ptitle])
ylabel('W 1')
axis([0 9 -1 6])
grid on

subplot(7,1,2)
plot(T(:), W2(:))
ylabel('W 2')
axis([0 9 -1 6])

```

grid on

```
subplot(7,1,3)
plot(T(:), W3(:))
ylabel('W 3')
axis([0 9 -1 6])
grid on
```

```
subplot(7,1,4)
plot(T(:), W4(:))
ylabel('W 4')
axis([0 9 -1 6])
grid on
```

```
subplot(7,1,5)
plot(T(:), W5(:), T(:), sW5(:), 'm--')
ylabel('W 5')
axis([0 9 -1 6])
grid on
```

```
subplot(7,1,6)
plot(T(:), W6(:))
ylabel('W 6')
axis([0 9 -1 6])
grid on
```

```
subplot(7,1,7)
plot(T(:), W7(:), T(:), sW7(:), 'm--')
ylabel('W 7')
xlabel('TIME FROM METERING FIX (min)')
axis([0 9 -1 6])
grid on
```


APPENDIX C

MERGING PROCEDURE PERFORMANCE SIMULATION

C.1 Overview

The FAST SL Merging Procedure Performance Simulation, which is illustrated in Figure C-1, is described in this appendix. The structure of this code is very similar to that of the Ordering Procedure Performance Simulation described in Appendix B. As a result, only the listings for those routines which are unique to the Merging Procedure Performance Simulation are included in this appendix.

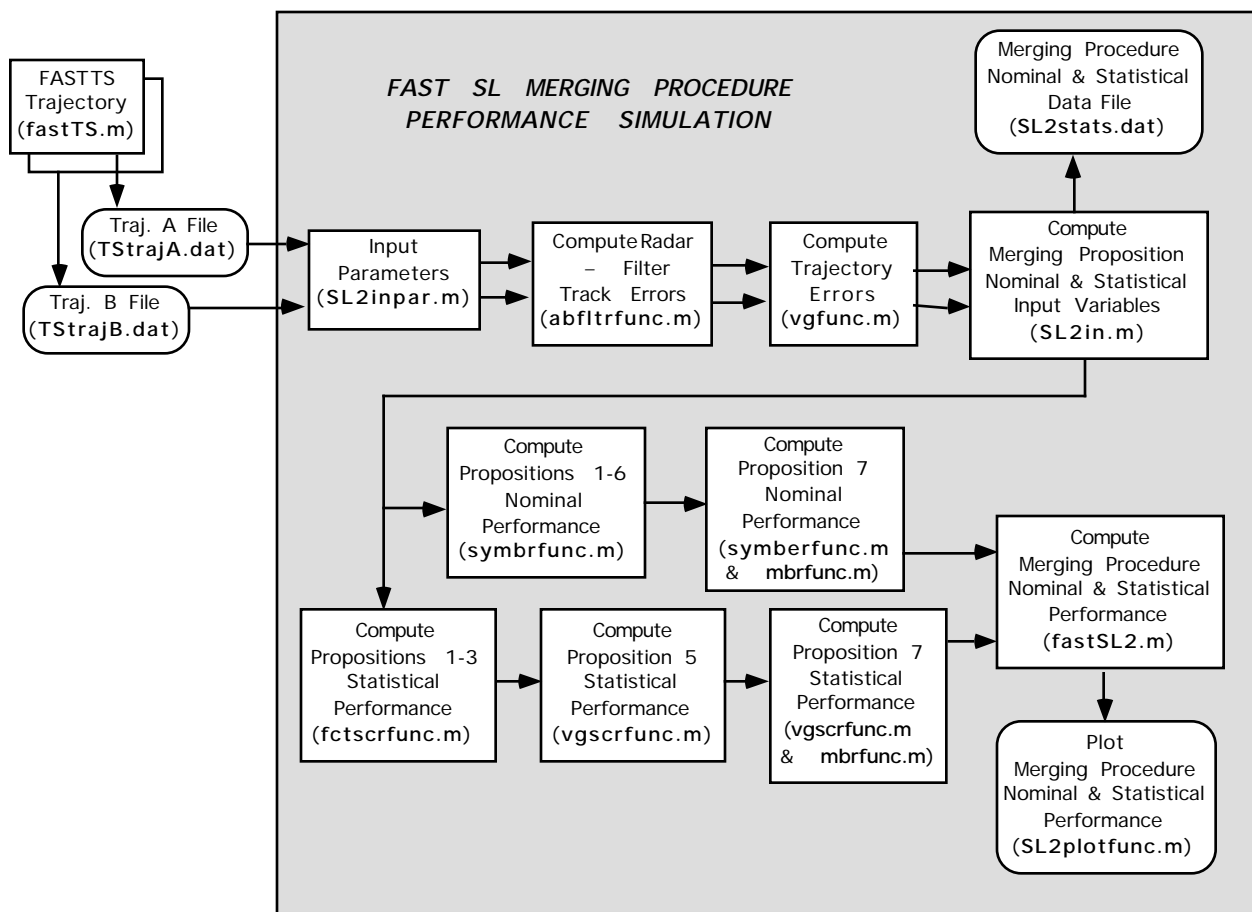


Figure C-1. FAST SL Merging Procedure Performance Simulation

The Merging Procedure Performance Simulation also starts out with two nominal aircraft trajectories. Since the focus of this simulation is on the merging of two aircraft from separate flight path segments, rather than the ordering of two in-track aircraft, a different set of aircraft would be used for this performance simulation.

The trajectories can be generated using the fastTS.m simulation (Appendix A) or can be obtained from some other source. The important consideration in the latter case is to format the trajectories in the same ASCII format used by fastTS.m.

The main code for this performance simulation is fastSL2.m, whose listing is presented in Section C.3. fastSL2.m obtains its input parameters and nominal trajectories through fastSL2inpar.m, whose listing is presented in Section C.4. Also needed in this script are any trajectory time biases to adjust the relative time or distance spacing between the two nominal aircraft trajectories.

Using these inputs, fastSL2in.m, whose listing is presented in Section C.5, computes the individual trajectory errors. It then computes the nominal and statistical input variables for use in the Merging Procedure Propositions found in fastSL2.m. Calculation of the trajectory errors is performed using the following MATLAB function scripts: intfunc.m (Section C.6), abfltrfunc.m (Section B.6), and vgfunc.m (Section B.7). The nominal and statistical input variables computed by fastSL2in.m is plotted using SL2inplotfunc.m, whose listing is presented in Section C.8.

With the nominal and statistical input variables from fastSL2in.m, fastSL2.m evaluates the seven Merging Procedure Propositions using both the nominal and statistical inputs (if these depend on ground speed errors). Similar to fastSL1.m, the first six nominal input Propositions use symbrfunc.m (Section B.9) while the seventh nominal input Proposition uses mbrfunc.m (Section B.11). The first three statistical input Propositions use fctscrfunc.m (Section C.7), while the fifth and seventh Proposition use vgscrfunc.m (Section B.10). In addition, the seventh Proposition also uses mbrfunc.m.

Using the nominal and statistical input Proposition results, fastSL2.m computes the Procedure results. These are plotted using SL2plotfunc.m (Section C.8).

C.2 Test Case

To illustrate the output obtained by executing fastSL2.m for a typical case, the case defined by fastSL1inpar.m (Section C.4) was executed. The resulting plots which were obtained are shown in Figures C-2 through C-16. Also shown in Table C-1 is an abstract of the statistical data output file, fastSL2stats.dat. The format used for this data file is summarized in Table C-2.

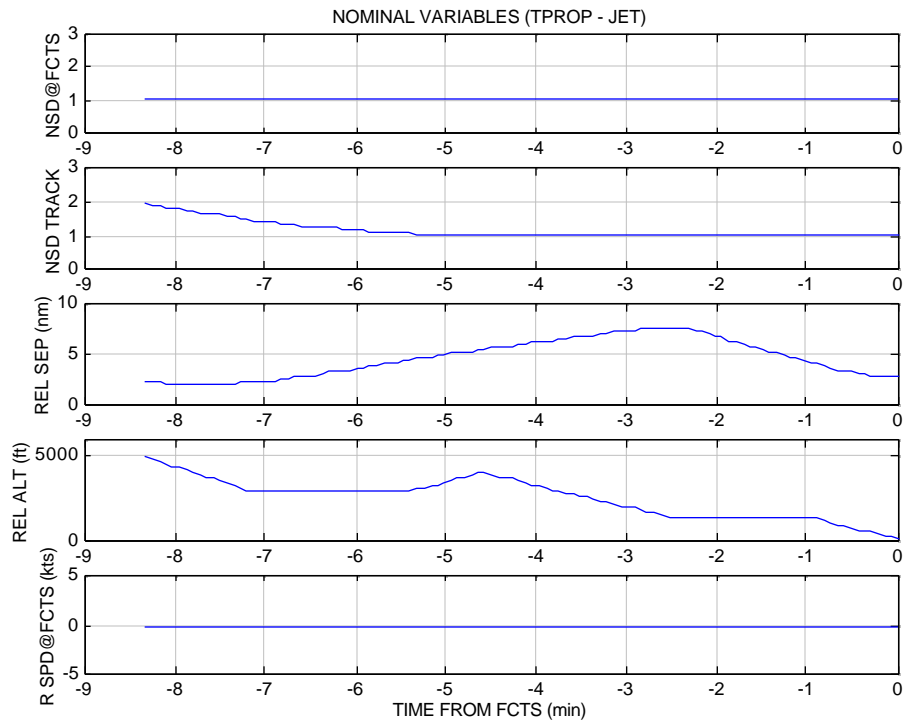


Figure C-2. Nominal Relative Variables (TProp - Jet)

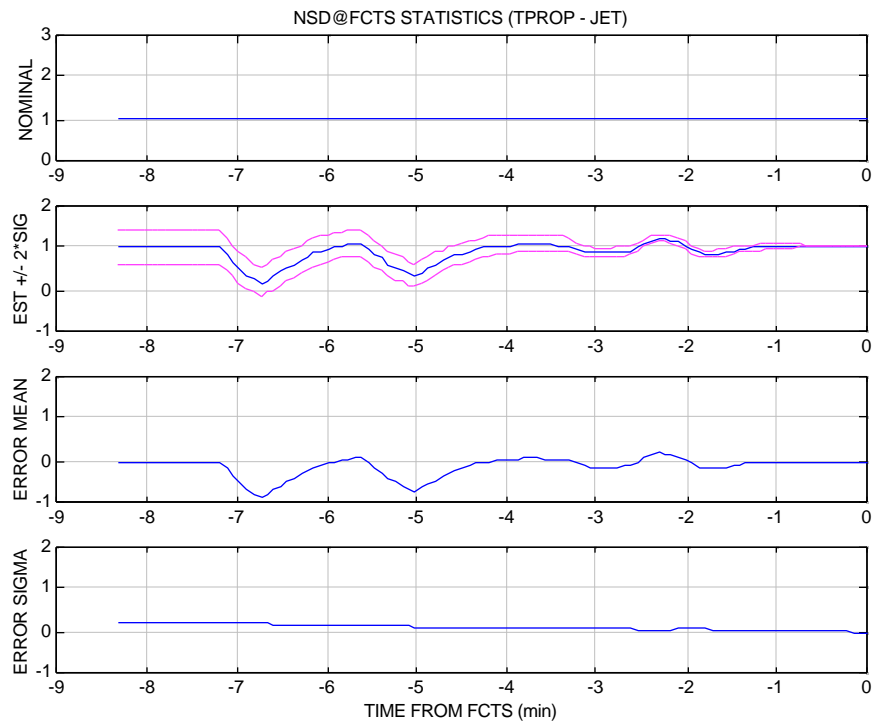


Figure C-3. NSD_{FCTS} Statistics (TProp - Jet)

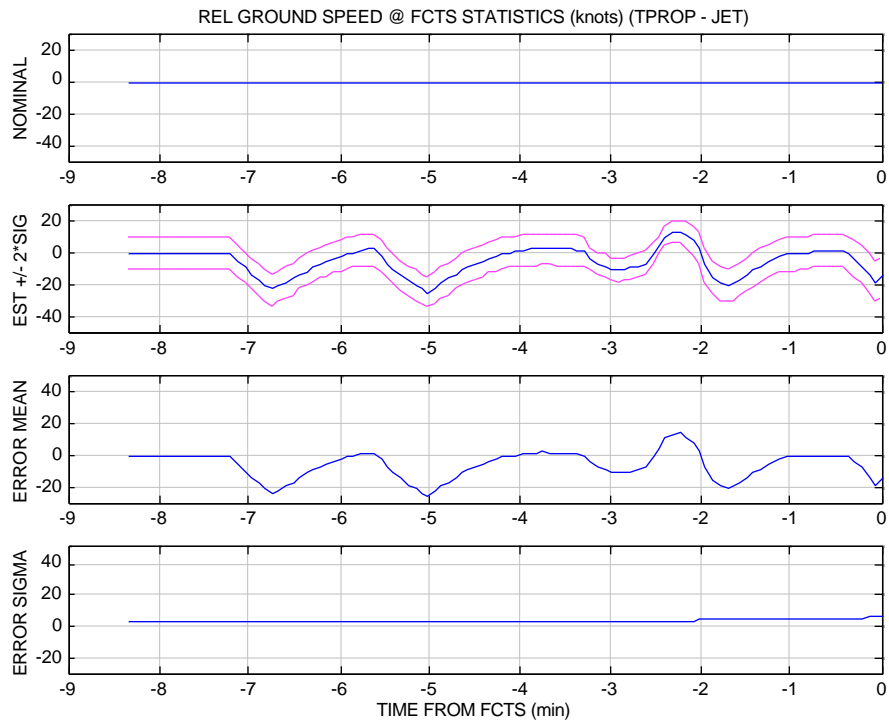


Figure C-4. Relative Ground Speed Statistics (TProp - Jet)

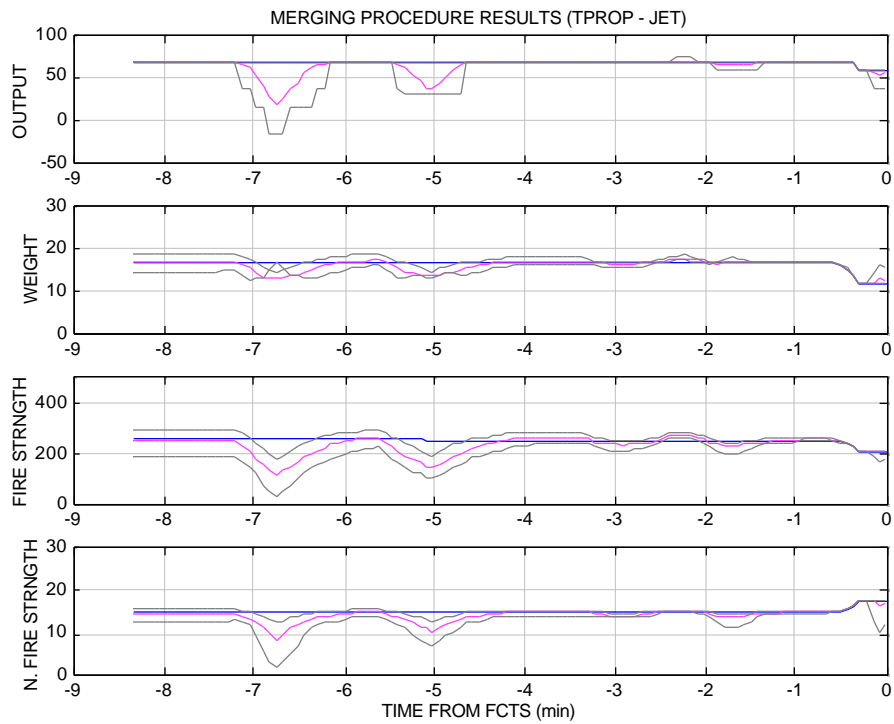


Figure C-5. Merging Procedure Nominal and Statistical Results (TProp - Jet)

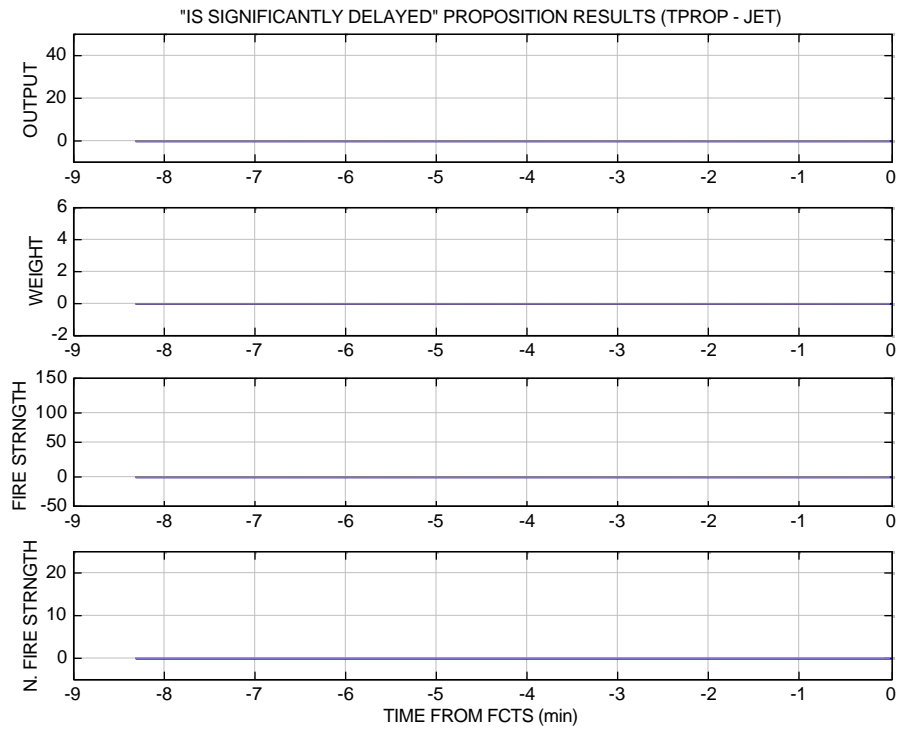


Figure C-6. 'Is Significantly Delayed' Proposition Nominal and Statistical Results (TProp - Jet)

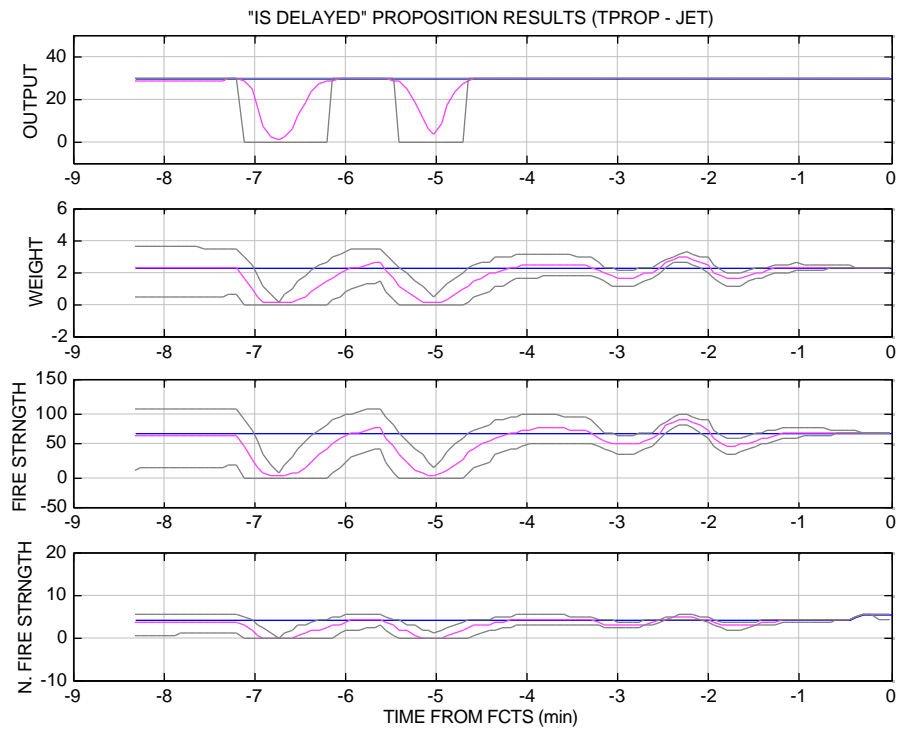


Figure C-7. 'Is Delayed' Proposition Nominal and Statistical Results (TProp - Jet)

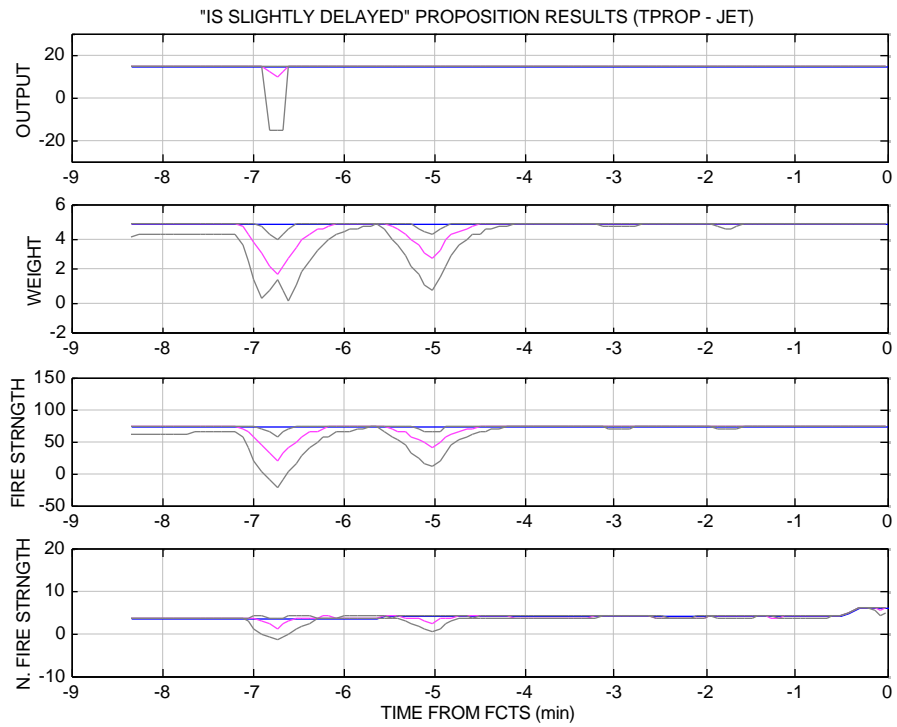


Figure C-8. 'Is Slightly Delayed' Proposition Nominal and Statistical Results (TProp - Jet)

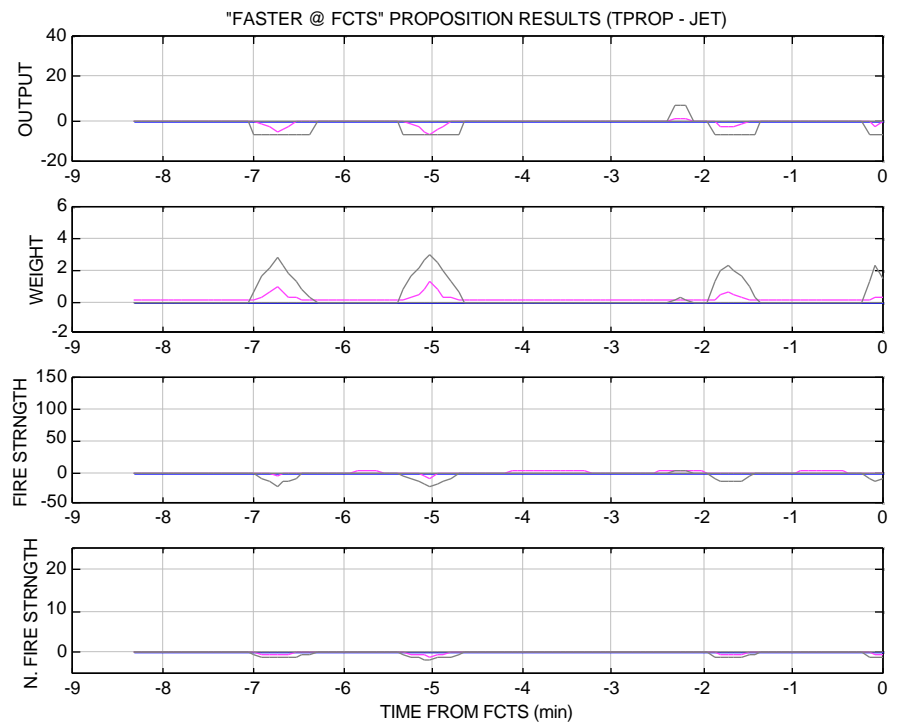


Figure C-9. 'Is Faster' Proposition Nominal and Statistical Results (TProp - Jet)

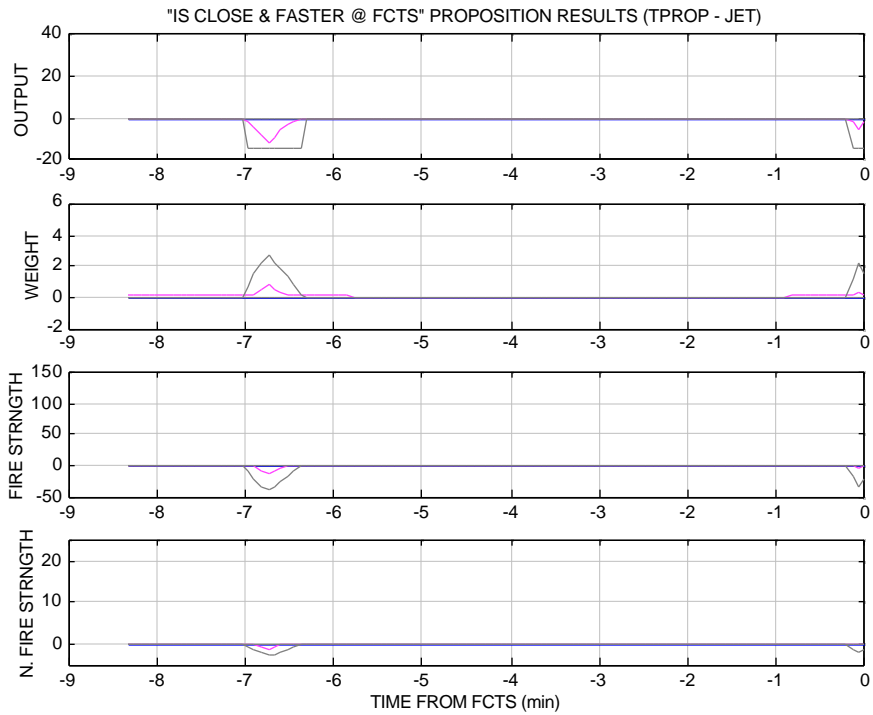


Figure C-10. 'Is Closer and Faster' Proposition Nominal and Statistical Results (TProp - Jet)

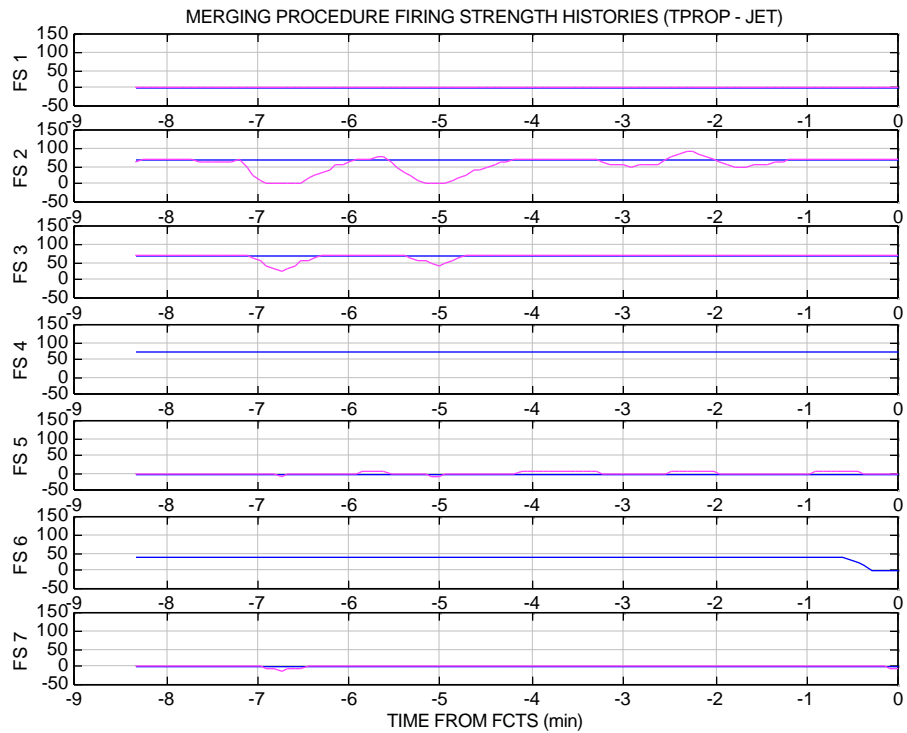


Figure C-11. Proposition Nominal and Statistical Firing Strength Results (TProp - Jet)

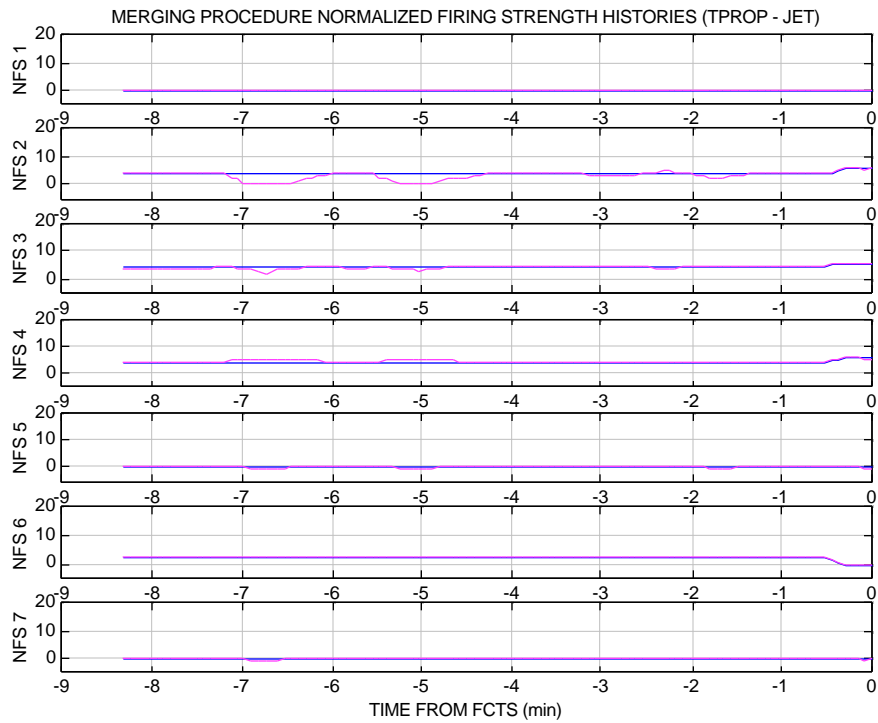


Figure C-12. Proposition Nominal and Statistical Normalized Firing Strength Results (TProp - Jet)

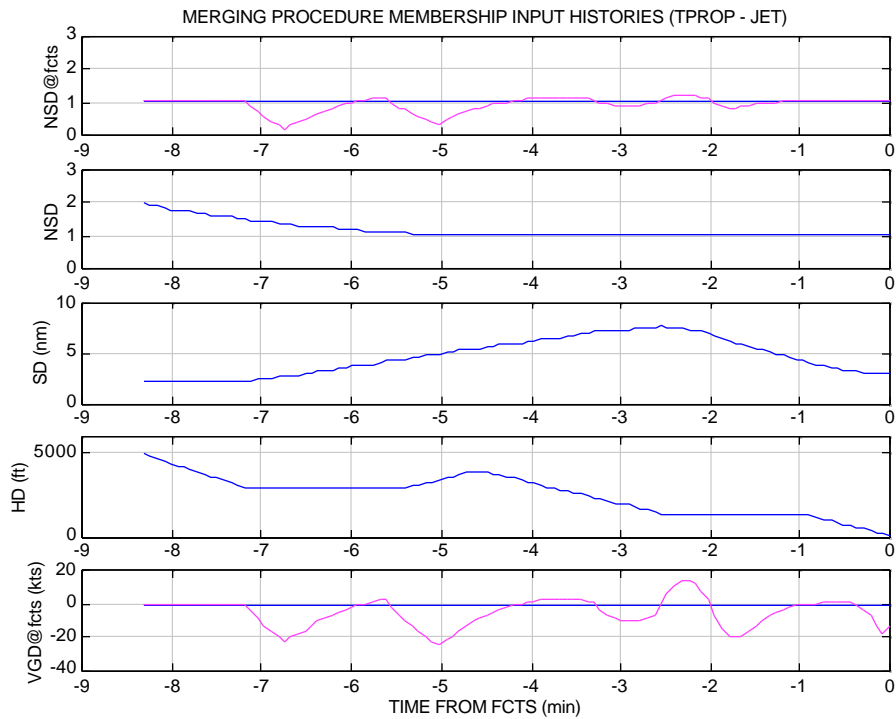


Figure C-13. Proposition Nominal and Statistical Input Variables (TProp - Jet)

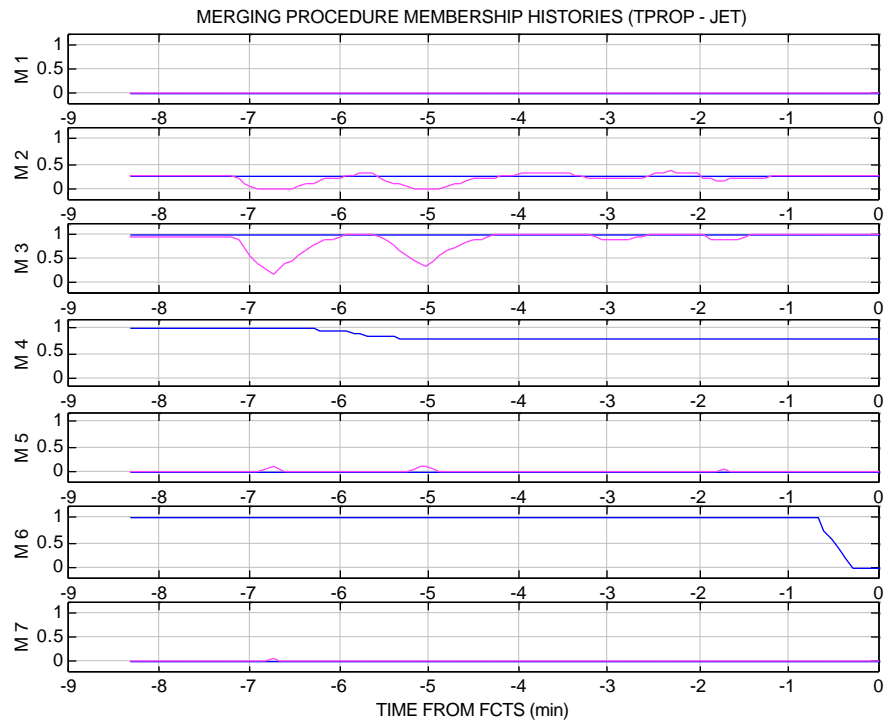


Figure C-14. Proposition Nominal and Statistical Membership Values (TProp - Jet)

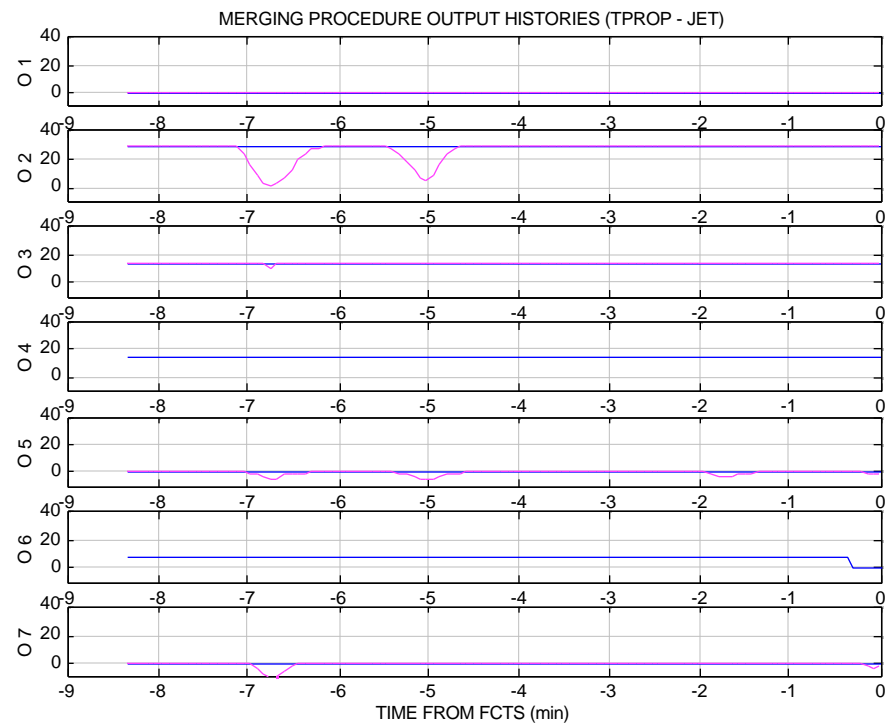


Figure C-15. Proposition Nominal and Statistical Output Values (TProp - Jet)

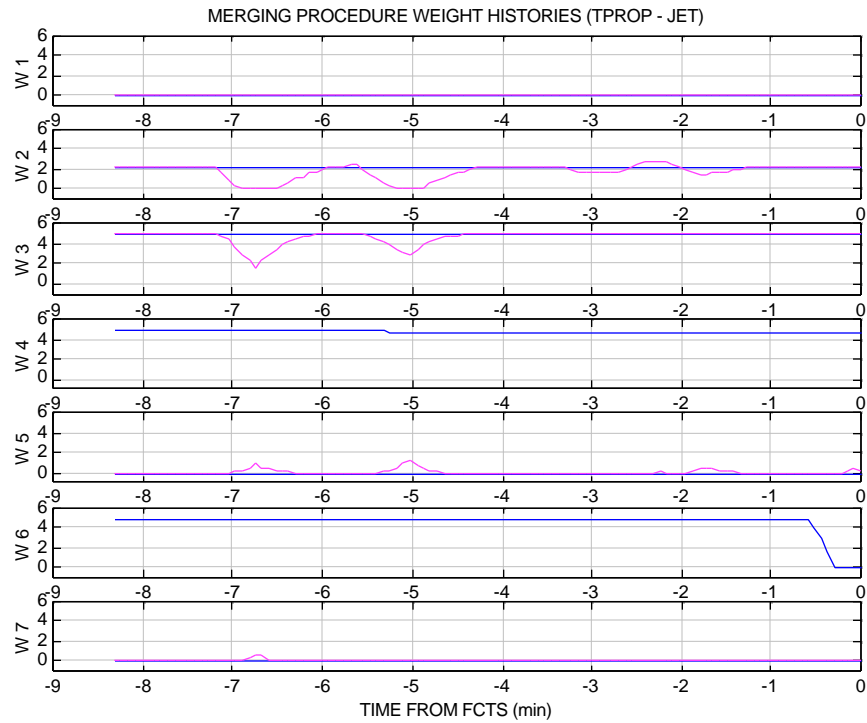


Figure C-16. Proposition Nominal and Statistical Weight (TProp - Jet)

Table C-1. fastSL2stats.dat Statistics Data File Abstract

| | | | | | |
|--------------|--------------|--------------|--------------|--------------|--------------|
| 0.000000e+00 | 1.940833e+00 | 2.362499e+00 | 5.010000e+03 | 3.103371e+00 | 6.851304e-02 |
| 6.513504e-01 | 0.000000e+00 | 4.937877e-01 | 4.694417e+00 | | |
| 7.500000e-02 | 1.911666e+00 | 2.300235e+00 | 4.879929e+03 | 3.103371e+00 | 6.272612e-02 |
| 6.498629e-01 | 0.000000e+00 | 4.561899e-01 | 4.726276e+00 | | |
| 1.500000e-01 | 1.882500e+00 | 2.244807e+00 | 4.749859e+03 | 3.103371e+00 | 5.974999e-02 |
| 6.447767e-01 | 0.000000e+00 | 4.385320e-01 | 4.732306e+00 | | |
| 2.250000e-01 | 1.853333e+00 | 2.196734e+00 | 4.619788e+03 | 3.103371e+00 | 5.902815e-02 |
| 6.426007e-01 | 0.000000e+00 | 4.372456e-01 | 4.760005e+00 | | |
| 3.000000e-01 | 1.824166e+00 | 2.156508e+00 | 4.489717e+03 | 3.103371e+00 | 5.985552e-02 |
| 6.409689e-01 | 0.000000e+00 | 4.475179e-01 | 4.792291e+00 | | |
| : | : | : | : | : | : |
| 8.100000e+00 | 1.034457e+00 | 3.040566e+00 | 3.983171e+02 | 3.103371e+00 | 2.798027e-02 |
| 2.223598e-02 | 0.000000e+00 | 7.461407e+00 | 5.929596e+00 | | |
| 8.175000e+00 | 1.034457e+00 | 3.020414e+00 | 3.007642e+02 | 3.103371e+00 | 3.232233e-02 |
| 1.514167e-02 | 0.000000e+00 | 1.292893e+01 | 6.056671e+00 | | |
| 8.250000e+00 | 1.034457e+00 | 3.012307e+00 | 2.032112e+02 | 3.103371e+00 | 2.221091e-02 |
| 7.659961e-03 | 0.000000e+00 | 1.776872e+01 | 6.127969e+00 | | |
| 8.325000e+00 | 1.034457e+00 | 3.010616e+00 | 1.056582e+02 | 3.103371e+00 | 0.000000e+00 |
| 0.000000e+00 | 0.000000e+00 | 1.383588e+01 | 6.145182e+00 | | |

Table C-2. fastSL2stats.dat File Format

| COLUMN | VARIABLE | UNITS |
|--------|---|-------|
| 1 | Time from Metering Fix | min |
| 2 | Nominal NSD Track | |
| 3 | Nominal Relative Separation | nm |
| 4 | Nominal Relative Altitude | ft |
| 5 | Nominal SD at FCTS | nm |
| 6 | Mean SD at FCTS | nm |
| 7 | Standard Deviation of SD at FCTS | nm |
| 8 | Nominal Relative Ground Speed | kts |
| 9 | Mean Relative Ground Speed | kts |
| 10 | Standard Deviation of Relative Ground Speed | kts |

C.3 fastSL2.m MATLAB Listing

```
% NASAFastSL2.m
% Simulation of FAST SL nominal fuzzy decision logic. It takes the ENU
% time-referenced aircraft trajectory for two aircraft following different
% paths to the same runway. By considering these as generic trajectories, multiple
% aircraft are simulated by time shifting the two input trajectories. The nominal
% trajectory variables are then used to evaluate the:
%
%      Merging Procedure of a GENERAL-Type Spatial Constraint
% -----

% developed by:      K. Tysen Mueller
%                   SEAGULL TECHNOLOGY, INC.
% developed on:      7 August 98
% modified on:       9 November 98

clear all
t0 = clock;

% ----- INPUTS -----

NASAFastSL2inpar; % Contains control constants, data, and trajectory files
% NOTE: User must start with fastSL1inpar.m before executing fasSL2in.m

NASAFastSL2in; % Calculation of FAST SL input errors based on the alpha-beta filter
% errors. It takes the ENU time-referenced aircraft trajectory and
% the radar error statistics and computes the time-referenced radar, ENU, time of
% arrival (ETA), and First Common Time Step (FCTS) estimation error mean and sigmas.
% Based on original alphabeta.m and on fastSLin.m simulations. Provides inputs for
% fastSL2.m.

%----- INITIALIZATION -----
```

D2R = pi/180; % deg to radians
 NM2FT = 6076.1033; % nautical miles to feet
 HR2S = 3600; % hours to seconds
 KTS2FPS = NM2FT/HR2S; % knots to feet/sec
 KF2F = 1000; % kilofeet to feet
 M2S = 60; % Minutes to seconds

% ----- MAIN LOOP -----

for k = 1:K

% ----- MERGING PROCEDURE of a GENERAL-Type (MPGT) Spatial Constraint -----

% MPGT propositions:

% O(k): Proposition 1 output value

% M(k) : Proposition 1 membership

% S(k) : Proposition 1 firing strength (same as weighted output)

% 'Significantly Ahead at FCTS' Membership, 'Significantly Favored' Consequent Functions:

[O1(k),M1(k),W1(k),S1(k)] = NASAsymbrfunc(NSDFCTS(k),2,4,45); %
 Nominal

[sO1(k),sM1(k),sW1(k),sS1(k)] = ...
 NASAfctscrfunc(SDFCTSeSt(k),sigSDFCTS(k),dsep12,dsep21,2,4,45); %
 Statistical

[sO1a(k),sM1a(k),sW1a(k),sS1a(k)] = NASAsymbrfunc(NSDFCTSeStM2S(k),2,4,45);
 % Min

[sO1b(k),sM1b(k),sW1b(k),sS1b(k)] = NASAsymbrfunc(NSDFCTSeStP2S(k),2,4,45);
 % Max

% 'Ahead at FCTS' Membership, 'Favored' Consequent Functions:

[O2(k),M2(k),W2(k),S2(k)] = NASAsymbrfunc(NSDFCTS(k),0.5,2.5,30); %
 Nominal

[sO2(k),sM2(k),sW2(k),sS2(k)] = ...
 NASAfctscrfunc(SDFCTSeSt(k),sigSDFCTS(k),dsep12,dsep21,0.5,2.5,30); %
 Statistical

[sO2a(k),sM2a(k),sW2a(k),sS2a(k)] = NASAsymbrfunc(NSDFCTSeStM2S(k),0.5,2.5,30);
 % Min

[sO2b(k),sM2b(k),sW2b(k),sS2b(k)] = NASAsymbrfunc(NSDFCTSeStP2S(k),0.5,2.5,30);
 % Max

% 'Slightly Ahead at FCTS' Membership, 'Slightly Favored' Consequent Functions:

[O3(k),M3(k),W3(k),S3(k)] = NASAsymbrfunc(NSDFCTS(k),0,1,15); %
Nominal

[sO3(k),sM3(k),sW3(k),sS3(k)] = ...
NASAfctscrfunc(SDFCTSe(k),sigSDFCTS(k),dsep12,dsep21,0,1,15); %
Statistical

[sO3a(k),sM3a(k),sW3a(k),sS3a(k)] = NASAsymbrfunc(NSDFCTSeM2S(k),0,1,15);
% Min

[sO3b(k),sM3b(k),sW3b(k),sS3b(k)] = NASAsymbrfunc(NSDFCTSeP2S(k),0,1,15);
% Max

% 'Ahead at Current Position' Membership, 'Slightly Favored' Consequent Functions:

[O4(k),M4(k),W4(k),S4(k)] = NASAsymbrfunc(NSD(k),0.25,1.25,15);

% 'Faster at FCTS' Membership, 'Marginally Favored' Consequent Functions:

[O5(k),M5(k),W5(k),S5(k)] = NASAsymbrfunc(dVG(k),20,60,7.5); %
Nominal

[sO5(k),sM5(k),sW5(k),sS5(k)] = NASAvgscrfunc(dVGest(k),sigdVG(k),20,60,7.5); %
Statistical

[sO5a(k),sM5a(k),sW5a(k),sS5a(k)] = NASAsymbrfunc(dVGestM2S(k),20,60,7.5); %
Min

[sO5b(k),sM5b(k),sW5b(k),sS5b(k)] = NASAsymbrfunc(dVGestP2S(k),20,60,7.5); %
Max

% 'Lower at Current Position' Membership, 'Marginally Favored' Consequent Functions:

[O6(k),M6(k),W6(k),S6(k)] = NASAsymbrfunc(dh(k),500,1000,7.5);

% Note that Proposition 7 uses an AND operation for the 'Close' and 'Faster'
% membership functions. This AND operation selects the output from these two
% membership functions for which there is minimum membership

% 'Close at Current Position' Membership, 'Slightly Favored' Consequent Functions:

[Out, Mbr, Wght, FS] = NASAmbrfunc(ds(k),0,1,4,0,15);

M7(k) = 0; O7(k) = 0; W7(k) = 0; S7(k) = 0;

% Nominal

[O7v(k),M7v(k),W7v(k),S7v(k)] = NASAsymbrfunc(dVG(k),20,60,15); % Nominal

if M7v(k) >= Mbr

```

M7(k) = Mbr;
Sgn = sign(dVG(k));
O7(k) = Out*Sgn;
W7(k) = Wght;
S7(k) = FS*Sgn;

elseif M7v(k) < Mbr

    M7(k) = M7v(k);
    O7(k) = O7v(k);
    W7(k) = W7v(k);
    S7(k) = S7v(k);

end

% Statistical

sM7(k) = 0; sO7(k) = 0; sW7(k) = 0; sS7(k) = 0;

[sO7v(k),sM7v(k),sW7v(k),sS7v(k)] = NASAvgscrfunc(dVGest(k),sigdVG(k),20,60,15); %
Statistical

if sM7v(k) >= Mbr

    sM7(k) = Mbr;
    sSgn = sign(dVGest(k));
    sO7(k) = Out*sSgn;
    sW7(k) = Wght;
    sS7(k) = FS*sSgn;

elseif sM7v(k) < Mbr

    sM7(k) = sM7v(k);
    sO7(k) = sO7v(k);
    sW7(k) = sW7v(k);
    sS7(k) = sS7v(k);

end

% Min

sM7a(k) = 0; sO7a(k) = 0; sW7a(k) = 0; sS7a(k) = 0;

[sO7va(k),sM7va(k),sW7va(k),sS7va(k)] = NASAasybrfunc(dVGestM2S(k),20,60,15); %
Min

if sM7va(k) >= Mbr

    sM7a(k) = Mbr;
    Sgn = sign(dVGestM2S(k));
    sO7a(k) = Out*Sgn;
    sW7a(k) = Wght;
    sS7a(k) = FS*Sgn;

```

```

elseif sM7va(k) < Mbr

    sM7a(k) = sM7va(k);
    sO7a(k) = sO7va(k);
    sW7a(k) = sW7va(k);
    sS7a(k) = sS7va(k);

end

% Max

sM7b(k) = 0; sO7b(k) = 0; sW7b(k) = 0; sS7b(k) = 0;

[sO7vb(k),sM7vb(k),sW7vb(k),sS7vb(k)] = NASAsymbrfunc(dVGestP2S(k),20,60,15); %
Max

if sM7vb(k) >= Mbr

    sM7b(k) = Mbr;
    Sgn = sign(dVGestP2S(k));
    sO7b(k) = Out*Sgn;
    sW7b(k) = Wght;
    sS7b(k) = FS*Sgn;

elseif sM7vb(k) < Mbr

    sM7b(k) = sM7vb(k);
    sO7b(k) = sO7vb(k);
    sW7b(k) = sW7vb(k);
    sS7b(k) = sS7vb(k);

end

% Compute nominal Procedure results:

OT(k) = O1(k) + O2(k) + O3(k) + O4(k) + O5(k) + O6(k) + O7(k); % Total output
WT(k) = W1(k) + W2(k) + W3(k) + W4(k) + W5(k) + W6(k) + W7(k); % Total weight
FST(k) = S1(k) + S2(k) + S3(k) + S4(k) + S5(k) + S6(k) + S7(k); % Total firing strength

if WT(k) == 0

    WT(k) = 1;

end

NFST(k) = FST(k)/WT(k); % Normalized total firing strength
NS1(k) = S1(k)/WT(k);
NS2(k) = S2(k)/WT(k);
NS3(k) = S3(k)/WT(k);
NS4(k) = S4(k)/WT(k);
NS5(k) = S5(k)/WT(k);
NS6(k) = S6(k)/WT(k);
NS7(k) = S7(k)/WT(k);

```

% Compute statistical Procedure results:

```

sOT(k) = sO1(k) + sO2(k) + sO3(k) + O4(k) + sO5(k) + O6(k) + sO7(k); % Total output
sWT(k) = sW1(k) + sW2(k) + sW3(k) + W4(k) + sW5(k) + W6(k) + sW7(k); % Total
weight
sFST(k) = sS1(k) + sS2(k) + sS3(k) + S4(k) + sS5(k) + S6(k) + sS7(k); % Total firing strength

if sWT(k) == 0

    sWT(k) = 1;

end

sNFST(k) = sFST(k)/sWT(k); % Normalized total firing strength
sNS1(k) = sS1(k)/sWT(k);
sNS2(k) = sS2(k)/sWT(k);
sNS3(k) = sS3(k)/sWT(k);
sNS4(k) = S4(k)/sWT(k);
sNS5(k) = sS5(k)/sWT(k);
sNS6(k) = S6(k)/sWT(k);
sNS7(k) = sS7(k)/sWT(k);

```

% Compute min results:

```

sOTa(k) = sO1a(k) + sO2a(k) + sO3a(k) + O4(k) + sO5a(k) + O6(k) + sO7a(k); % Total
output
sWTa(k) = sW1a(k) + sW2a(k) + sW3a(k) + W4(k) + sW5a(k) + W6(k) + sW7a(k); %
Total weight
sFSTa(k) = sS1a(k) + sS2a(k) + sS3a(k) + S4(k) + sS5a(k) + S6(k) + sS7a(k); % Total firing
strength

if sWTa(k) == 0

    sWTa(k) = 1;

end

sNFSTa(k) = sFSTa(k)/sWTa(k); % Normalized total firing strength
sNS1a(k) = sS1a(k)/sWTa(k);
sNS2a(k) = sS2a(k)/sWTa(k);
sNS3a(k) = sS3a(k)/sWTa(k);
sNS4a(k) = S4(k)/sWTa(k);
sNS5a(k) = sS5a(k)/sWTa(k);
sNS6a(k) = S6(k)/sWTa(k);
sNS7a(k) = sS7a(k)/sWTa(k);

```

% Compute max results:

```

sOTb(k) = sO1b(k) + sO2b(k) + sO3b(k) + O4(k) + sO5b(k) + O6(k) + sO7b(k); % Total
output
sWTb(k) = sW1b(k) + sW2b(k) + sW3b(k) + W4(k) + sW5b(k) + W6(k) + sW7b(k); %
Total weight

```

```

sFSTb(k) = sS1b(k) + sS2b(k) + sS3b(k) + S4(k) + sS5b(k) + S6(k) + sS7b(k); % Total firing
strength

if sWTb(k) == 0

    sWTb(k) = 1;

end

sNFSTb(k) = sFSTb(k)/sWTb(k);           % Normalized total firing strength
sNS1b(k) = sS1b(k)/sWTb(k);
sNS2b(k) = sS2b(k)/sWTb(k);
sNS3b(k) = sS3b(k)/sWTb(k);
sNS4b(k) = S4(k)/sWTb(k);
sNS5b(k) = sS5b(k)/sWTb(k);
sNS6b(k) = S6(k)/sWTb(k);
sNS7b(k) = sS7b(k)/sWTb(k);

end

% ----- PLOTS -----

NASASL2plotfunc(ptitle,t,NSDFCTS,NSDFCTSest,NSDFCTSestM2S,NSDFCTSestP2S,muNS
DFCTS, ...
    sigNSDFCTS,NSD,ds,dh,dVG,dVGest,dVGestM2S,dVGestP2S,mudVG,sigdVG,
...
    M1,M2,M3,M4,M5,M6,M7,OT,O1,O2,O3,O4,O5,O6,O7,WT,W1,W2,W3, ...
    W4,W5,W6,W7,FST,S1,S2,S3,S4,S5,S6,S7,NFST,NS1,NS2,NS3,NS4, ...
    NS5,NS6,NS7,sM1,sM2,sM3,sM5,sM7,sOT,sO1,sO2,sO3,sO5,sO7, ...
    sWT,sW1,sW2,sW3,sW5,sW7,sFST,sS1,sS2,sS3,sS5,sS7,sNFST, ...
    sNS1,sNS2,sNS3,sNS4,sNS5,sNS6,sNS7,sM1a,sM2a,sM3a,sM5a,sM7a, ...
    sOTa,sO1a,sO2a,sO3a,sO5a,sO7a,sWTa,sW1a,sW2a,sW3a,sW5a,sW7a, ...
    sFSTa,sS1a,sS2a,sS3a,sS5a,sS7a,sNFSTa,sNS1a,sNS2a,sNS3a,sNS4a, ...
    sNS5a,sNS6a,sNS7a,sM1b,sM2b,sM3b,sM5b,sM7b,sOTb,sO1b,sO2b,sO3b, ...
    sO5b,sO7b,sWTb,sW1b,sW2b,sW3b,sW5b,sW7b,sFSTb,sS1b,sS2b,sS3b, ...
    sS5b,sS7b,sNFSTb,sNS1b,sNS2b,sNS3b,sNS4b,sNS5b,sNS6b,sNS7b);

Run_time = etime(clock,t0)

```

C.4 fastSL2inpar.m MATLAB Listing

```

% NASAFastSL2inpar.m
% Sets up the input parameters, trajectories, and data bases used by fastSL1in.m

% Developed by:          K.Tysen Mueller
%                      SEAGULL TECHNOLOGY, INC.
% Developed on:         30 September 98
% Modified on:          19 October 98

% ***** INPUTS *****

```

Seagull Technology, Inc. Proprietary Information

```

ptitle = '(TPROP - JET)';          % Common plot title for trajectory pair

%sigr = 0.0625;          % Range sigma (nmi) -- Ed Horvath/FAATC
sigr = 0.00823;          % Range sigma (nmi) --George Hunter/CTAS Filter Study
sigz = 0.16;             % Azimuth sigma (deg) -- both refs.

t01= 0;                  % Actual start time for 1st aircraft trajectory (must be multiple
                        % of delt; sec)
t02= 0;                  % Actual start time for 2nd aircraft trajectory (must be multiple
                        % of delt; sec)
tf1 = 1026.0;            % Actual final time for 1st aircraft trajectory (must be multiple
                        % of delt; sec)
tf2 = 1017.0;            % Actual final time for 2nd aircraft trajectory time (must be
                        % multiple of delt; sec)

dt = 0.5;                % fastTS.m trajectory time interval (sec)
delt = 4.5;              % Radar sweep interval (sec) -- Note: while the actual
                        % sweep period has been reported as 4.6 or 4.7 secs,
                        % the use of an integer multiple value of (dt) is more
                        % convenient
dtout = 4.5;             % Output time interval (must be an integer multiple
                        % of delt. Note: all the input times listed below have been
                        % adjusted such that they are integer multiples of delt).

tbias1 = 40.5;           % Additive time bias to 1st trajectory time history (must
                        % be multiple of delt; sec) Note: with value shown first
                        % aircraft will be ahead of second aircraft by 3 nm at FCTS
tbias2 = 0;              % Additive time bias to 2nd trajectory time history (must
                        % be multiple of delt; sec)

tFCTS = 540;             % First common time step (FCTS) for 1st and 2nd aircraft.
                        % Referenced to unbiased time scales (must be multiple
                        % of delt; sec)
tbiasFCTS = -40.5;       % Additive time bias to adjust nominal tFCTS (must be multiple
                        % of delt; sec)

sRE1 = 57.56;            % Distance for 1st aircraft to Runway Edge from Metering Fix (nm)
sRE2 = 61.02;            % Distance for 2nd aircraft to Runway Edge from Metering Fix (nm)

dsep12 = 3.0;            % Min spacing if 1st aircraft is ahead of 2nd (nm)
dsep21 = 4.0;            % Min spacing if 2nd aircraft is ahead of 1st (nm)

xR = 29.0;               % TRACON radar east location (nmi) -- origin is SW metering fix.
yR = 23.4;               % TRACON radar north location (nmi)
zR = 0;                  % TRACON radar altitude (kft)

rmax = 17;               % Max number of successful radar sweeps to reach range a-b
                        % filter gain steady-state values
amax = 7;                % Max number of successful radar sweeps to reach azimuth a-b
                        % filter gain steady-state values

% ***** Load TS trajectory *****

```

% Trajectories have a time interval of dt

load fastTSprop.dat
fastraj1 = fastTSprop; % 1st trajectory

load fastTSjet.dat
fastraj2 = fastTSjet; % 2nd trajectory

% ***** Alpha-Beta Tracking Filter Gains *****

% If there are no false correlations, the gain history is as follows:

FAB = [14 64 64 64 64; 17 54 43 54 43; 22 44 24 44 24; 24 38 16 38 16; ...
26 34 12 36 14; 30 30 9 36 14; 32 27 7 34 12; 34 24 6 34 12; ...
36 22 5 34 12; 40 20 4 34 12; 41 19 3 34 12; 42 18 3 34 12; ...
43 17 3 34 12; 44 16 2 34 12; 45 15 2 34 12; 46 14 2 34 12; ...
47 13 2 34 12];

C.5 fastSL2in.m MATLAB Listing

% **NASAFastSL2in.m**

% Simulation of FAST SL input errors based on the alpha-beta filter errors. It
% takes the ENU time-referenced aircraft trajectory and the radar error statistics
% and computes the time-referenced radar, ENU, time of arrival (ETA), and First
% Common Time Step (FCTS) estimation error mean and sigmas. Based on original
% alphabeta.m and on fastSLin.m simulations. Provides inputs for fastSL2.m.

% developed by: K. Tysen Mueller
% SEAGULL TECHNOLOGY, INC.
% developed on: 30 September 98
% modified on: 9 November 98

%----- INITIALIZATION -----

D2R = pi/180; % deg to radians
NM2FT = 6076.1033; % nautical miles to feet
HR2S = 3600; % hours to seconds
KTS2FPS = NM2FT/HR2S; % knots to feet/sec
KF2F = 1000; % kilofeet to feet
NM2M = 6076.1033/3.28084; % Nautical miles to meters
M2S = 60; % Minutes to seconds
SXTY42D = 1/64; % 64'th to decimal

% Assuming that both trajectories have the same time origin (e.g: time from metering fix):
% Compute adjusted output start time for both trajectories:
t01Out = (t01 + tbias1);
t02Out = (t02 + tbias2);
tstart = [t01Out,t02Out];

% Compute adjusted final output time for both trajectories:
% Note: Since for the Merging Procedure everything pivots around FCTS, tFCTS is selected

```

%      as the final time.
TFCTS = (tFCTS + tbiasFCTS);          % Effective time of tFCTS (sec)
tf1Out = (TFCTS + tbias1);
tf2Out = (TFCTS + tbias2);
tfinal = [tf1Out,tf2Out];

% Compute common adjusted output times:
T0 = min(tstart);                    % Common initial output time (sec)
TF = min(tfinal);                    % Common final output time (sec)

dk = (dtout/dt);
k1 = (t01Out/dt + 1 - dk);           % Start index for 1st trajectory file
k2 = (t02Out/dt + 1 - dk);           % Start index for 2nd trajectory file

k01 = t01Out/dtout;                  % Start index for 1st trajectory output data
k02 = t02Out/dtout;                  % Start index for 2nd trajectory output data

K = (TF - T0)/dtout + 1;             % Number of common time steps
kFCTS1 = k01 + K;                    % Final index for 1st trajectory input data
kFCTS2 = k02 + K;                    % Final index for 2nd trajectory input data

ndt1 = ((tf1 - t01)/dtout + 1);      % Number of output time points for 1st trajectory
K1 = ((tf1 - t01)/dt + 1);           % Number of 1st trajectory points (used by intfunc.m)
L1 = ((tf1Out - t01Out)/dt + 1);     % Number of 1st trajectory points to tFCTS

ndt2 = ((tf2 - t02)/dtout + 1);      % Number of output time points for 2nd trajectory
K2 = ((tf2 - t02)/dt + 1);           % Number of 2nd trajectory points (used by intfunc.m)
L2 = ((tf2Out - t02Out)/dt + 1);     % Number of 2nd trajectory points to tFCTS

rcorrlim = (2*sigr*NM2FT)/(delt^2); % a-b filter absolute value of range acceleration
% correlation limit (ft/s^2) --- computed based on
% maximum range deviation allowed from predicted
% aircraft track, between consecutive radar sweeps

acorrlim = (2*sigz*D2R)/(delt^2);    % a-b filter absolute value of azimuth acceleration
% correlation limit (deg/s^2) --- computed based on
% maximum azimuth deviation allowed from predicted
% aircraft track, between consecutive radar sweeps

% The basic trajectories are unpacked at dtout rather than at dt. Any time biases are
% applied during the relative aircraft variable calculations

for k = 1:ndt1                       % 1st trajectory @ dtout time interval
    kk = (k-1)*(dtout/dt) + 1;
    t1(k) = (fastraj1(kk,1));         % Time (sec)
    x1(k) = fastraj1(kk,2);           % East position (ft)
    y1(k) = fastraj1(kk,3);           % North position (ft)
    z1(k) = fastraj1(kk,4);           % Altitude (ft)
    vx1(k) = fastraj1(kk,5);          % East velocity (ft/sec)
    vy1(k) = fastraj1(kk,6);          % North velocity (ft/sec)
    vz1(k) = fastraj1(kk,7);          % Altitude rate (ft/sec)
    s1(k) = fastraj1(kk,8);           % Distance traveled (ft)
    psi1(k) = fastraj1(kk,9)*D2R;     % Heading, psig (rads)
    sdot1(k) = fastraj1(kk,10);       % Ground speed, Vg, (ft/sec)

```

```

    psidot1(k) = fastraj1(kk,11)*D2R;      % Heading rate (rads/sec)
    sdbldt1(k) = fastraj1(kk,12);         % Ground accel (ft/sec^2)
end

for k = 1:ndt2                            % 2nd trajectory @ dtout time interval
    kk = (k-1)*(dtout/dt) + 1;
    t2(k) = (fastraj2(kk,1));             % Time (sec)
    x2(k) = fastraj2(kk,2);               % East position (ft)
    y2(k) = fastraj2(kk,3);               % North position (ft)
    z2(k) = fastraj2(kk,4);               % Altitude (ft)
    vx2(k) = fastraj2(kk,5);              % East velocity (ft/sec)
    vy2(k) = fastraj2(kk,6);              % North velocity (ft/sec)
    vz2(k) = fastraj2(kk,7);              % Altitude rate (ft/sec)
    s2(k) = fastraj2(kk,8);                % Distance traveled (ft)
    psi2(k) = fastraj2(kk,9)*D2R;          % Heading, psig (rads)
    sdot2(k) = fastraj2(kk,10);            % Ground speed, Vg, (ft/sec)
    psidot2(k) = fastraj2(kk,11)*D2R;      % Heading rate (rads/sec)
    sdbldt2(k) = fastraj2(kk,12);          % Ground accel (ft/sec^2)
end

% Compute inverse ground speed integral history (used in NSD@FCTS calculation)
% Note: Integral is taken from runway edge to current location/time @ dt

[INTFCTS1,INT1,Integ1] = NASAintfunc(fastraj1(:,10),fastraj1(:,12),dt,L1,K1);
[INTFCTS2,INT2,Integ2] = NASAintfunc(fastraj2(:,10),fastraj2(:,12),dt,L2,K2);

% Initialize the alpha-beta filter initial and noise covariance matrices

Cm = zeros(2,2);
Cm(1,1) = (sigr*NM2FT)^2;
Cm(2,2) = (sigz*D2R)^2;

HPh1 = zeros(4,4,ndt1); HPh2 = zeros(4,4,ndt2);
HmuV1 = zeros(4,ndt1); HmuV2 = zeros(4,ndt2);

% ***** 1st Aircraft Trajectory Tracking Errors *****

% Radar estimation error calculations:

[muRN1,muEta1,PrN1] = NASAabfltrfunc(ndt1,delt,x1,y1,z1,vx1,vy1,vz1,psi1, ...
    sdot1,sdbldt1,psidot1,FAB,xR,yR,zR,rcorrlim,acorrlim,rmax,amax,Cm);

% Ground speed, heading, and path estimation error calculations:

[muVG1,muPsiG1,muS1,sigVG1,sigPsiG1,sigS1,HmuV1,HPh1] = ...
    NASAvgfunc(ndt1,dt,dtout,x1,y1,z1,xR,yR,zR,psi1,sdot1,muRN1,muEta1,PrN1);

% ***** 2nd Aircraft Trajectory Tracking Errors *****

% Radar estimation error calculations:

[muRN2,muEta2,PrN2] = NASAabfltrfunc(ndt2,delt,x2,y2,z2,vx2,vy2,vz2,psi2, ...
    sdot2,sdbldt2,psidot2,FAB,xR,yR,zR,rcorrlim,acorrlim,rmax,amax,Cm);

```

% Ground speed, heading and path estimation error calculations:

```
[muVG2,muPsiG2,muS2,sigVG2,sigPsiG2,sigS2,HmuV2,HPh2] = ...
    NASAvgfunc(ndt2,dt,dtout,x2,y2,z2,xR,yR,zR,psi2,sdot2,muRN2,muEta2,PrN2);
```

% ----- AIRCRAFT RELATIVE VARIABLE CALCULATIONS -----

% The relative aircraft variable calculations are performed by shifting the
 % time index, but not the times, of the respective two trajectories. For the 3
 % aircraft, the tFCTS needs to be shifted by tbiasP1 since the time at which P1
 % reaches the downwind segment, determines the effective tFCTS.

% ***** Aircraft 1 ahead of 2 *****

```
T = zeros(1,K); t = zeros(1,K);
SDFCTS = zeros(1,K); SDFCTSeST = zeros(1,K);
muSDFCTS = zeros(1,K); sigSDFCTS = zeros(1,K);
NSDFCTS = zeros(1,K); NSDFCTSeST = zeros(1,K);
muNSDFCTS = zeros(1,K); sigNSDFCTS = zeros(1,K);
dVG = zeros(1,K); mudVG = zeros(1,K); sigdVG = zeros(1,K);
```

% Compute relative aircraft statistics data:

```
[T,dVG,mudVG,sigdVG,SDFCTS,muSDFCTS,sigSDFCTS] = NASAreIvfunc(T0,TF,dt,dtout, ...
    k01,k02,K,kFCTS1,kFCTS2,Integ1,Integ2,muVG1,muVG2,sigVG1,sigVG2, ...
    s1,s2,sdot1,sdot2,HmuV1,HmuV2,HPh1,HPh2,sRE1,sRE2);
```

% Nominal relative aircraft trajectory data calculations:

for k = 1:K

```
t(k) = T(k) - (TF/M2S);           % Time relative to tFCTS (min)
k1 = k01 + k;
k2 = k02 + k;
```

% NSD Track:

```
S1(k) = sRE1 - (s1(k1)/NM2FT);
S2(k) = sRE2 - (s2(k2)/NM2FT);
SD(k) = (S2(k) - S1(k));           % SD Track wrt RE (nm)
```

```
if SD(k) > 0
    NSD(k) = SD(k)/dsep12;          % NSD Track wrt RE
else
    NSD(k) = SD(k)/dsep21;
end
```

% NSD @ FCTS:

```
if SDFCTS(k) > 0
```

```

    NSDFCTS(k) = SDFCTS(k)/dsep12;           % NSD @ FCTS wrt RE
else
    NSDFCTS(k) = SDFCTS(k)/dsep21;
end

% Relative altitude (ft):

dh(k) = z2(k2) - z1(k1);

% Line-of-sight separation distance (nm):

ds(k) = (sqrt((x1(k1) - x2(k2))^2 + (y1(k1) - y2(k2))^2))/NM2FT;

% Statistical calculations:

% SD @ FCTS estimate wrt RE (nm):

SDFCTSeSt(k) = SDFCTS(k) + muSDFCTS(k);           % Estimate
SDFCTSeStM2S(k) = (SDFCTSeSt(k) - 2*sigSDFCTS(k)); % Estimate - 2*sigma
SDFCTSeStP2S(k) = (SDFCTSeSt(k) + 2*sigSDFCTS(k)); % Estimate + 2*sigma

% NSD @ FCTS statistics:

if SDFCTSeSt(k) > 0

    NSDFCTSeSt(k) = SDFCTSeSt(k)/dsep12;
    muNSDFCTS(k) = muSDFCTS(k)/dsep12;
    sigNSDFCTS(k) = sigSDFCTS(k)/dsep12;

else

    NSDFCTSeSt(k) = SDFCTSeSt(k)/dsep21;
    muNSDFCTS(k) = muSDFCTS(k)/dsep21;
    sigNSDFCTS(k) = sigSDFCTS(k)/dsep21;

end

if SDFCTSeStM2S(k) > 0
    NSDFCTSeStM2S(k) = SDFCTSeStM2S(k)/dsep12;           % NSD @ FCTS - 2*sigma
else
    NSDFCTSeStM2S(k) = SDFCTSeStM2S(k)/dsep21;
end

if SDFCTSeStP2S(k) > 0
    NSDFCTSeStP2S(k) = SDFCTSeStP2S(k)/dsep12;           % NSD @ FCTS + 2*sigma
else
    NSDFCTSeStP2S(k) = SDFCTSeStP2S(k)/dsep21;
end

% Relative ground speed @ FCTS estimate (kts)

dVGest(k) = (dVG(k) + mudVG(k));           % Estimate
dVGestM2S(k) = dVGest(k) - 2*sigdVG(k);     % Estimate - 2*sigma

```

```

dVGestP2S(k) = dVGest(k) + 2*sigdVG(k);           % Estimate + 2*sigma

end

% Output statistics:

fastSL2stats = zeros(K,10);

fastSL2stats(:,1) = T(:);           % Time from metering fix (min)
fastSL2stats(:,2) = NSD(:);         % Nominal NSD Track
fastSL2stats(:,3) = ds(:);          % Nominal relative separation (nm)
fastSL2stats(:,4) = dh(:);          % Nominal relative altitude (ft)
fastSL2stats(:,5) = SDFCTS(:);       % SD @ FCTS (nm)
fastSL2stats(:,6) = muSDFCTS(:);     % SD @ FCTS estimation error mean (nm)
fastSL2stats(:,7) = sigSDFCTS(:);    % SD @ FCTS estimation error sigma (nm)
fastSL2stats(:,8) = dVG(:);          % Relative ground speed (kts)
fastSL2stats(:,9) = mudVG(:);        % Relative ground speed mean (kts)
fastSL2stats(:,10) = sigdVG(:);      % Relative ground speed sigma (kts)

% ----- OUTPUT FILES -----

save fastSL2stats.datfastSL2stats -ascii

```

C.6 intfunc.m MATLAB Listing

% **NASAintfunc.m**

```

% Input: All variables in @ dt
%   speed = ground speed (ft/sec)
%   accel = ground accel (ft/sec^2)
%   dt   = input & output time interval (sec)
%   L    = number of trajectory time points to tFCTS
%   K    = number of trajectory time points

% Output: All variables out @ dt
%   INTFCTS = time integral of inverse ground speed wrt to FCTS (sec^2/ft)
%   INT     = time integral of inverse ground speed wrt to runway edge (sec^2/ft)
%   Integ   = time integral of inverse ground speed wrt to metering fix (sec^2/ft)

% Note: These integrals are used in SD @ FCTS calculations. For these calculations
%   the ground speed and acceleration are referenced to an arbitrary downwind
%   reference point, positive toward the metering fix. Hence, these integrals
%   would have to have their polarity reversed to be consistent with this
%   definition.

function[INTFCTS,INT,Integ] = NASAintfunc(speed,accel,dt,L,K)

Integ(1) = 0;

```

```

for k = 2:K
    if accel(k) == 0
        DInteg = dt/speed(k);
    else
        DInteg = (1/accel(k))*log(abs((speed(k) + dt*accel(k))/speed(k)));
    end
    Integ(k) = Integ(k-1) + DInteg;    % Referenced to initial position
end

for j = 1:K
    INT(j) = Integ(K) - Integ(j);    % Referenced to final position (runway edge)
end
for j = 1:L
    INTFCTS(j) = Integ(L) - Integ(j); % Referenced to tFCTS
end

```

C.7 fctscrfunc.m MATLAB Listing

```

% NASAfctscrfunc.m
% Computes the NSD @ FCTS mean membership, output value, weight, and firing strength
% (score). The calculations presented below can be performed in the NSD @ FCTS domain
% or in the SD @ FCTS domain. Both will yield the same answers if the membership
% function is scaled correctly. The latter approach was used, for convenience, to
% simplify the difficulty in handling the difference between the positive and negative
% input value cases.

% Assumption:  1) NSD @ FCTS membership function is symmetric about 0
%              2) SD @ FCTS membership function doesn't have to be symmetric about 0
%              3) for NSD @ FCTS, a and b are positive with a > b
%              4) SD @ FCTS membership is zero between -aBA and +aAB (See below)
%              5) SD @ FCTS membership is 1 for < -bBA and for > +bAB

% Input:  SDFCTSeST,sigSDFCTS = SD @ FCTS estimate and sigma (nm)
%         dsepAB,dsepBA = minimum required separation distance for aircraft A ahead of B
%         (AB) and for the reverse order.
%         a,b = NSD @ FCTS trapezoidal membership function is described by the start of
%         the dead band (a) and the point at which full membership is reached (b)
%         c = non-zero consequent output value

% Output: NSDfOUT = consequent function output value
%         NSDfMBR = membership value
%         NSDfWGHT = consequent function weight
%         NSDfSCR = consequent function firing strength

function[NSDfOUT,NSDfMBR,NSDfWGHT,NSDfSCR] = ...
    NASAfctscrfunc(SDFCTSeST,sigSDFCTS,dsepAB,dsepBA,a,b,c)

% Initialize outputs
NSDfOUT = 0;
NSDfMBR = 0;

```

```

NSDfWGHT = 0;
NSDfSCR = 0;
pNSDfa = 0;
pNSDfb = 0;
SQRT2 = sqrt(2);

% Compute SD@FCTS equivalent membership function parameters
aBA = a*dsepBA;
bBA = b*dsepBA;
aAB = a*dsepAB;
bAB = b*dsepAB;

% Determine if probability density function is very narrow (peaked):
% In the calculations below, a check is performed to determine if the probability
% density function is so narrow (small sigSDFCTS) that the result is equivalent to
% using SDFCTSeST in the nominal proposition calculations. In addition, where the
% probability density function is wide enough, the numerical integral calculations
% will integrate only over the region defined by -/+3*sigSDFCTS about SDFCTSeST.

if sigSDFCTS > 0.1

% Decision Error Probabilities:

    pNSDfa = (0.5)*(erfc((aBA + SDFCTSeST)/(SQRT2*sigSDFCTS)));
    pNSDfb = (0.5)*(erfc((aAB - SDFCTSeST)/(SQRT2*sigSDFCTS)));

% Expected Output Values:

    NSDfaout = -c*pNSDfa;           % Case a output value
    NSDfbout = c*pNSDfb;           % Case b output value
    NSDfOUT = NSDfbout + NSDfaout; % Case total output value

% Expected Membership and Weight:

    v = 0;
    dvBA = (bBA - aBA)/200;        % integration step size
    dvAB = (bAB - aAB)/200;        %

    NSDfaMBR = 0;
    NSDfbMBR = 0;

    NSDfaWGHT = 0;
    NSDfbWGHT = 0;

    CBAM = 1/((sqrt(2*pi))*sigSDFCTS*(bBA - aBA));
    CABM = 1/((sqrt(2*pi))*sigSDFCTS*(bAB - aAB));

    CBAW = -5/((sqrt(2*pi))*sigSDFCTS*(bBA - aBA)^2);
    CABW = -5/((sqrt(2*pi))*sigSDFCTS*(bAB - aAB)^2);

    dnm = (2*sigSDFCTS^2);

    if (SDFCTSeST - 3*sigSDFCTS) <= -aBA % Check if -3 sigma pdf tail to left of -aBA

```

```

for n = -(bBA-dvBA):dvBA:-(aBA)      % BA membership & weight integral

    v = n;
    zsM = (v + aBA);
    zsW = (v + aBA)*(v - aBA + 2*bBA);

    ua = ((v - SDFCTsest)^2)/dnm;
    Expnta = exp(-ua);

    NSDfaMBR = NSDfaMBR + dvBA*CBAM*zsM*Expnta;
    NSDfaWGHT = NSDfaWGHT + dvBA*CBAW*zsW*Expnta;

end

ExpntaF = 0.5*erfc((bBA + SDFCTsest)/(SQRT2*sigSDFCTS));

NSDfaMBR = NSDfaMBR + ExpntaF;          % Case a mean membership
NSDfaWGHT = NSDfaWGHT + 5*ExpntaF;     % Case a mean weight

end

if (SDFCTsest + 3*sigSDFCTS) >= aAB % Check if +3 sigma pdf tail to right of +aAB

    for n = (aAB):dvAB:(bAB-dvAB)      % AB membership & weight integral

        v = n;
        zsM = (v - aAB);
        zsW = (v - aAB)*(v + aAB - 2*bAB);

        ub = ((v - SDFCTsest)^2)/dnm;
        Expntb = exp(-ub);

        NSDfbMBR = NSDfbMBR + dvAB*CABM*zsM*Expntb;
        NSDfbWGHT = NSDfbWGHT + dvAB*CABW*zsW*Expntb;

    end

    ExpntbF = 0.5*erfc((bAB - SDFCTsest)/(SQRT2*sigSDFCTS));

    NSDfbMBR = NSDfbMBR + ExpntbF;      % Case b mean membership
    NSDfbWGHT = NSDfbWGHT + 5*ExpntbF;  % Case b mean weight

end

NSDfMBR = NSDfaMBR + NSDfbMBR;          % Mean membership
NSDfWGHT = NSDfaWGHT + NSDfbWGHT;      % Mean weight

% Expected Firing Strength (Score):

NSDfaSCR = -c*NSDfaWGHT;
NSDfbSCR = c*NSDfbWGHT;
NSDfSCR = NSDfaSCR + NSDfbSCR;

else

```

```
% If sigSDFCTS is too small, the expected value integrals, above, behave
% like a Dirac delta function whose output is only determined by the estimate,
% SDFCTSeest. If no numerical integration were required, then this situation
% would be handled automatically by the above logic.
```

```
    if SDFCTSeest >= 0

        [NSDFOUT,NSDFMBR,NSDFWGHT,NSDFSCR] =
NASAmbrfunc(SDFCTSeest,aAB,0,bAB,1,c);

    else

        [NSDFOUT,NSDFMBR,NSDFWGHT,NSDFSCR] = NASAmbrfunc(SDFCTSeest,-aBA,0,-
bBA,1,-c);

    end

end
```

C.8 SL2plotfunc.m MATLAB Listing

```
% NASASL2plotfunc.m
% Plots relative statistics computed by fastSL2.m for two aircraft

% Created by:      K. Tysen Mueller
%                  SEAGULL TECHNOLOGY, INC.
% Created on:      6 October 98
% Modified on:     21 October 98

function[] = NASASL2plotfunc(ptitle,T,NSDFCTS,NSDFCTSeest,NSDFCTSeestM2S, ...
    NSDFCTSeestP2S,muNSDFCTS,sigNSDFCTS, ...
    NSD,ds,dh,dVG,dVGest,dVGestM2S,dVGestP2S,mudVG,sigdVG, ...
    M1,M2,M3,M4,M5,M6,M7,OT,O1,O2,O3,O4,O5,O6,O7,WT,W1,W2,W3, ...
    W4,W5,W6,W7,FST,S1,S2,S3,S4,S5,S6,S7,NFST,NS1,NS2,NS3,NS4, ...
    NS5,NS6,NS7,sM1,sM2,sM3,sM5,sM7,sOT,sO1,sO2,sO3,sO5,sO7, ...
    sWT,sW1,sW2,sW3,sW5,sW7,sFST,sS1,sS2,sS3,sS5,sS7,sNFST, ...
    sNS1,sNS2,sNS3,sNS4,sNS5,sNS6,sNS7,sM1a,sM2a,sM3a,sM5a,sM7a, ...
    sOTa,sO1a,sO2a,sO3a,sO5a,sO7a,sW1a,sW2a,sW3a,sW5a,sW7a, ...
    sFSTa,sS1a,sS2a,sS3a,sS5a,sS7a,sNFSTa,sNS1a,sNS2a,sNS3a, ...
    sNS4a,sNS5a,sNS6a,sNS7a,sM1b,sM2b,sM3b,sM5b,sM7b, ...
    sOTb,sO1b,sO2b,sO3b,sO5b,sO7b,sWTb,sW1b,sW2b,sW3b,sW5b,sW7b, ...
    sFSTb,sS1b,sS2b,sS3b,sS5b,sS7b,sNFSTb,sNS1b,sNS2b,sNS3b,sNS4b, ...
    sNS5b,sNS6b,sNS7b);

subplot(5,1,1)
plot(T(:), NSDFCTS(:),'b')
title(['NOMINAL VARIABLES ',ptitle])
ylabel('NSD@FCTS')
axis([-9 0 0 3])
grid on
```

```
subplot(5,1,2)
plot(T(:), NSD(:),'b')
ylabel('NSD TRACK')
axis([-9 0 0 3])
grid on
```

```
subplot(5,1,3)
plot(T(:), ds(:),'b')
ylabel('REL SEP (nm)')
axis([-9 0 0 10])
grid on
```

```
subplot(5,1,4)
plot(T(:), dh(:),'b')
ylabel('REL ALT (ft)')
axis([-9 0 0 6000])
grid on
```

```
subplot(5,1,5)
plot(T(:), dVG(:),'b')
ylabel('R SPD@FCTS (kts)')
xlabel('TIME FROM FCTS (min)')
axis([-9 0 -5 5])
grid on
```

figure

```
subplot(4,1,1)
plot(T(:), NSDFCTS(:),'b')
title(['NSD@FCTS STATISTICS ',ptitle])
ylabel('NOMINAL')
axis([-9 0 0 3])
grid on
```

```
subplot(4,1,2)
plot(T(:), NSDFCTSeST(:),'b',T(:), NSDFCTSeSTM2S(:),'--m', ...
      T(:), NSDFCTSeSTP2S(:),'--m')
ylabel('EST +/- 2*SIG')
axis([-9 0 -1 2])
grid on
```

```
subplot(4,1,3)
plot(T(:), muNSDFCTS(:),'b')
ylabel('ERROR MEAN')
axis([-9 0 -1 2])
grid on
```

```
subplot(4,1,4)
plot(T(:), sigNSDFCTS(:),'b')
ylabel('ERROR SIGMA')
xlabel('TIME FROM FCTS (min)')
axis([-9 0 -1 2])
grid on
```

figure

```
subplot(4,1,1)
plot(T(:), dVG(:),'b')
title(['REL GROUND SPEED @ FCTS STATISTICS (knots) ',ptitle])
ylabel('NOMINAL')
axis([-9 0 -50 30])
grid on
```

```
subplot(4,1,2)
plot(T(:), dVGest(:),'b',T(:), dVGestM2S(:),'--m', ...
      T(:), dVGestP2S(:),'--m');
ylabel('EST +/- 2*SIG')
axis([-9 0 -50 30])
grid on
```

```
subplot(4,1,3)
plot(T(:), mudVG(:),'b')
ylabel('ERROR MEAN')
axis([-9 0 -30 50])
grid on
```

```
subplot(4,1,4)
plot(T(:), sigdVG(:),'b')
ylabel('ERROR SIGMA')
xlabel('TIME FROM FCTS (min)')
axis([-9 0 -30 50])
grid on
```

figure

```
subplot(4,1,1)
plot(T(:), OT(:),T(:), sOT(:),'m--',T(:), sOTa(:),'k-.',T(:), sOTb(:),'k-.')
title(['MERGING PROCEDURE RESULTS ',ptitle])
ylabel('OUTPUT')
axis([-9 0 -50 100])
grid on
```

```
subplot(4,1,2)
plot(T(:), WT(:),T(:), sWT(:),'m--',T(:), sWTa(:),'k-.',T(:), sWTb(:),'k-.')
ylabel('WEIGHT')
axis([-9 0 0 30])
grid on
```

```
subplot(4,1,3)
plot(T(:), FST(:),T(:), sFST(:),'m--',T(:), sFSTa(:),'k-.',T(:), sFSTb(:),'k-.')
ylabel('FIRE STRNGTH')
axis([-9 0 0 500])
grid on
```

```
subplot(4,1,4)
plot(T(:), NFST(:),T(:), sNFST(:),'m--',T(:), sNFSTa(:),'k-.',T(:), sNFSTb(:),'k-.')
ylabel('N. FIRE STRNGTH')
```

```
xlabel('TIME FROM FCTS (min)')
axis([-9 0 0 30])
grid on
```

figure

```
subplot(4,1,1)
plot(T(:), O1(:),T(:), sO1(:),'m--',T(:), sO1a(:),'k-',T(:), sO1b(:),'k-')
title(['"IS SIGNIFICANTLY DELAYED" PROPOSITION RESULTS ',ptitle])
ylabel('OUTPUT')
axis([-9 0 -10 50])
grid on
```

```
subplot(4,1,2)
plot(T(:), W1(:),T(:), sW1(:),'m--',T(:), sW1a(:),'k-',T(:), sW1b(:),'k-')
ylabel('WEIGHT')
axis([-9 0 -2 6])
grid on
```

```
subplot(4,1,3)
plot(T(:), S1(:),T(:), sS1(:),'m--',T(:), sS1a(:),'k-',T(:), sS1b(:),'k-')
ylabel('FIRE STRNGTH')
axis([-9 0 -50 150])
grid on
```

```
subplot(4,1,4)
plot(T(:), NS1(:),T(:), sNS1(:),'m--',T(:), sNS1a(:),'k-',T(:), sNS1b(:),'k-')
ylabel('N. FIRE STRNGTH')
xlabel('TIME FROM FCTS (min)')
axis([-9 0 -5 25])
grid on
```

figure

```
subplot(4,1,1)
plot(T(:), O2(:),T(:), sO2(:),'m--',T(:), sO2a(:),'k-',T(:), sO2b(:),'k-')
title(['"IS DELAYED" PROPOSITION RESULTS ',ptitle])
ylabel('OUTPUT')
axis([-9 0 -10 50])
grid on
```

```
subplot(4,1,2)
plot(T(:), W2(:),T(:), sW2(:),'m--',T(:), sW2a(:),'k-',T(:), sW2b(:),'k-')
ylabel('WEIGHT')
axis([-9 0 -2 6])
grid on
```

```
subplot(4,1,3)
plot(T(:), S2(:),T(:), sS2(:),'m--',T(:), sS2a(:),'k-',T(:), sS2b(:),'k-')
ylabel('FIRE STRNGTH')
axis([-9 0 -50 150])
grid on
```

```
subplot(4,1,4)
```

```
plot(T(:), NS2(:),T(:), sNS2(:),'m--',T(:), sNS2a(:),'k-',T(:), sNS2b(:),'k-.')
ylabel('N. FIRE STRNGTH')
xlabel('TIME FROM FCTS (min)')
axis([-9 0 -10 20])
grid on
```

figure

```
subplot(4,1,1)
plot(T(:), O3(:),T(:), sO3(:),'m--',T(:), sO3a(:),'k-',T(:), sO3b(:),'k-.')
title(['"IS SLIGHTLY DELAYED" PROPOSITION RESULTS ',ptitle])
ylabel('OUTPUT')
axis([-9 0 -30 30])
grid on
```

```
subplot(4,1,2)
plot(T(:), W3(:),T(:), sW3(:),'m--',T(:), sW3a(:),'k-',T(:), sW3b(:),'k-.')
ylabel('WEIGHT')
axis([-9 0 -2 6])
grid on
```

```
subplot(4,1,3)
plot(T(:), S3(:),T(:), sS3(:),'m--',T(:), sS3a(:),'k-',T(:), sS3b(:),'k-.')
ylabel('FIRE STRNGTH')
axis([-9 0 -50 150])
grid on
```

```
subplot(4,1,4)
plot(T(:), NS3(:),T(:), sNS3(:),'m--',T(:), sNS3a(:),'k-',T(:), sNS3b(:),'k-.')
ylabel('N. FIRE STRNGTH')
xlabel('TIME FROM FCTS (min)')
axis([-9 0 -10 20])
grid on
```

figure

```
subplot(4,1,1)
plot(T(:), O5(:),T(:), sO5(:),'m--',T(:), sO5a(:),'k-',T(:), sO5b(:),'k-.')
title(['"FASTER @ FCTS" PROPOSITION RESULTS ',ptitle])
ylabel('OUTPUT')
axis([-9 0 -20 40])
grid on
```

```
subplot(4,1,2)
plot(T(:), W5(:),T(:), sW5(:),'m--',T(:), sW5a(:),'k-',T(:), sW5b(:),'k-.')
ylabel('WEIGHT')
axis([-9 0 -2 6])
grid on
```

```
subplot(4,1,3)
plot(T(:), S5(:),T(:), sS5(:),'m--',T(:), sS5a(:),'k-',T(:), sS5b(:),'k-.')
ylabel('FIRE STRNGTH')
axis([-9 0 -50 150])
grid on
```

```
subplot(4,1,4)
plot(T(:), NS5(:),T(:), sNS5(:),'m--',T(:), sNS5a(:),'k-',T(:), sNS5b(:),'k-.')
ylabel('N. FIRE STRNGTH')
xlabel('TIME FROM FCTS (min)')
axis([-9 0 -5 25])
grid on
```

figure

```
subplot(4,1,1)
plot(T(:), O7(:),T(:), sO7(:),'m--',T(:), sO7a(:),'k-',T(:), sO7b(:),'k-.')
title(['"IS CLOSE & FASTER @ FCTS" PROPOSITION RESULTS ',ptitle])
ylabel('OUTPUT')
axis([-9 0 -20 40])
grid on
```

```
subplot(4,1,2)
plot(T(:), W7(:),T(:), sW7(:),'m--',T(:), sW7a(:),'k-',T(:), sW7b(:),'k-.')
ylabel('WEIGHT')
axis([-9 0 -2 6])
grid on
```

```
subplot(4,1,3)
plot(T(:), S7(:),T(:), sS7(:),'m--',T(:), sS7a(:),'k-',T(:), sS7b(:),'k-.')
ylabel('FIRE STRNGTH')
axis([-9 0 -50 150])
grid on
```

```
subplot(4,1,4)
plot(T(:), NS7(:),T(:), sNS7(:),'m--',T(:), sNS7a(:),'k-',T(:), sNS7b(:),'k-.')
ylabel('N. FIRE STRNGTH')
xlabel('TIME FROM FCTS (min)')
axis([-9 0 -5 25])
grid on
```

figure

```
subplot(7,1,1)
plot(T(:), S1(:),T(:), sS1(:),'m--')
title(['MERGING PROCEDURE FIRING STRENGTH HISTORIES ',ptitle])
ylabel('FS 1')
axis([-9 0 -50 150])
grid on
```

```
subplot(7,1,2)
plot(T(:), S2(:),T(:), sS2(:),'m--')
ylabel('FS 2')
axis([-9 0 -50 150])
grid on
```

```
subplot(7,1,3)
plot(T(:), S3(:),T(:), sS3(:),'m--')
ylabel('FS 3')
```

```

axis([-9 0 -50 150])
grid on

subplot(7,1,4)
plot(T(:), S4(:))
ylabel('FS 4')
axis([-9 0 -50 150])
grid on

subplot(7,1,5)
plot(T(:), S5(:),T(:), sS5(:),'m--')
ylabel('FS 5')
axis([-9 0 -50 150])
grid on

subplot(7,1,6)
plot(T(:), S6(:))
ylabel('FS 6')
axis([-9 0 -50 150])
grid on

subplot(7,1,7)
plot(T(:), S7(:),T(:), sS7(:),'m--')
ylabel('FS 7')
xlabel('TIME FROM FCTS (min)')
axis([-9 0 -50 150])
grid on

figure

subplot(7,1,1)
plot(T(:), NS1(:),T(:), sNS1(:), 'm--')
title(['MERGING PROCEDURE NORMALIZED FIRING STRENGTH HISTORIES ',ptitle])
ylabel('NFS 1')
axis([-9 0 -5 20])
grid on

subplot(7,1,2)
plot(T(:), NS2(:),T(:), sNS2(:), 'm--')
ylabel('NFS 2')
axis([-9 0 -5 20])
grid on

subplot(7,1,3)
plot(T(:), NS3(:),T(:), sNS3(:), 'm--')
ylabel('NFS 3')
axis([-9 0 -5 20])
grid on

subplot(7,1,4)
plot(T(:),NS4(:),T(:), sNS4(:), 'm--')
ylabel('NFS 4')
axis([-9 0 -5 20])
grid on

```

```
subplot(7,1,5)
plot(T(:), NS5(:),T(:), sNS5(:), 'm--')
ylabel('NFS 5')
axis([-9 0 -5 20])
grid on
```

```
subplot(7,1,6)
plot(T(:), NS6(:),T(:), sNS6(:), 'm--')
ylabel('NFS 6')
axis([-9 0 -5 20])
grid on
```

```
subplot(7,1,7)
plot(T(:), NS7(:),T(:), sNS7(:), 'm--')
ylabel('NFS 7')
xlabel('TIME FROM FCTS (min)')
axis([-9 0 -5 20])
grid on
```

figure

```
subplot(5,1,1)
plot(T(:), NSDFCTS(:),T(:), NSDFCTSet(:),'m--')
title(['MERGING PROCEDURE MEMBERSHIP INPUT HISTORIES ',ptitle])
ylabel('NSD@fcts')
axis([-9 0 0 3])
grid on
```

```
subplot(5,1,2)
plot(T(:), NSD(:))
ylabel('NSD')
axis([-9 0 0 3])
grid on
```

```
subplot(5,1,3)
plot(T(:), ds(:))
ylabel('SD (nm)')
axis([-9 0 0 10])
grid on
```

```
subplot(5,1,4)
plot(T(:), dh(:))
ylabel('HD (ft)')
axis([-9 0 0 6000])
grid on
```

```
subplot(5,1,5)
plot(T(:), dVG(:),T(:), dVGest(:),'m--')
ylabel('VGD@fcts (kts)')
xlabel('TIME FROM FCTS (min)')
axis([-9 0 -40 20])
grid on
```

figure

```
subplot(7,1,1)
plot(T(:), M1(:),T(:), sM1(:),'m--')
title(['MERGING PROCEDURE MEMBERSHIP HISTORIES ',ptitle])
ylabel('M 1')
axis([-9 0 -0.2 1.2])
grid on
```

```
subplot(7,1,2)
plot(T(:), M2(:),T(:), sM2(:),'m--')
ylabel('M 2')
axis([-9 0 -0.2 1.2])
grid on
```

```
subplot(7,1,3)
plot(T(:), M3(:),T(:), sM3(:),'m--')
ylabel('M 3')
axis([-9 0 -0.2 1.2])
grid on
```

```
subplot(7,1,4)
plot(T(:), M4(:))
ylabel('M 4')
axis([-9 0 -0.2 1.2])
grid on
```

```
subplot(7,1,5)
plot(T(:), M5(:),T(:), sM5(:),'m--')
ylabel('M 5')
axis([-9 0 -0.2 1.2])
grid on
```

```
subplot(7,1,6)
plot(T(:), M6(:))
ylabel('M 6')
axis([-9 0 -0.2 1.2])
grid on
```

```
subplot(7,1,7)
plot(T(:), M7(:),T(:), sM7(:),'m--')
ylabel('M 7')
xlabel('TIME FROM FCTS (min)')
axis([-9 0 -0.2 1.2])
grid on
```

figure

```
subplot(7,1,1)
plot(T(:), O1(:),T(:), sO1(:),'m--')
title(['MERGING PROCEDURE OUTPUT HISTORIES ',ptitle])
ylabel('O 1')
axis([-9 0 -10 40])
grid on
```

```
subplot(7,1,2)
plot(T(:), O2(:),T(:), sO2(:),'m--')
ylabel('O 2')
axis([-9 0 -10 40])
grid on
```

```
subplot(7,1,3)
plot(T(:), O3(:),T(:), sO3(:),'m--')
ylabel('O 3')
axis([-9 0 -10 40])
grid on
```

```
subplot(7,1,4)
plot(T(:), O4(:))
ylabel('O 4')
axis([-9 0 -10 40])
grid on
```

```
subplot(7,1,5)
plot(T(:), O5(:),T(:), sO5(:),'m--')
ylabel('O 5')
axis([-9 0 -10 40])
grid on
```

```
subplot(7,1,6)
plot(T(:), O6(:))
ylabel('O 6')
axis([-9 0 -10 40])
grid on
```

```
subplot(7,1,7)
plot(T(:), O7(:),T(:), sO7(:),'m--')
ylabel('O 7')
xlabel('TIME FROM FCTS (min)')
axis([-9 0 -10 40])
grid on
```

figure

```
subplot(7,1,1)
plot(T(:), W1(:),T(:), sW1(:),'m--')
title(['MERGING PROCEDURE WEIGHT HISTORIES ',ptitle])
ylabel('W 1')
axis([-9 0 -1 6])
grid on
```

```
subplot(7,1,2)
plot(T(:), W2(:),T(:), sW2(:),'m--')
ylabel('W 2')
axis([-9 0 -1 6])
grid on
```

```
subplot(7,1,3)
```

```
plot(T(:), W3(:),T(:), sW3(:),'m--')
ylabel('W 3')
axis([-9 0 -1 6])
grid on
```

```
subplot(7,1,4)
plot(T(:), W4(:))
ylabel('W 4')
axis([-9 0 -1 6])
grid on
```

```
subplot(7,1,5)
plot(T(:), W5(:),T(:), sW5(:),'m--')
ylabel('W 5')
axis([-9 0 -1 6])
grid on
```

```
subplot(7,1,6)
plot(T(:), W6(:))
ylabel('W 6')
axis([-9 0 -1 6])
grid on
```

```
subplot(7,1,7)
plot(T(:), W7(:),T(:), sW7(:),'m--')
ylabel('W 7')
xlabel('TIME FROM FCTS (min)')
axis([-9 0 -1 6])
grid on
```